O I P E JC87
OCT 15 2001
PATENT & TRADEMARK OFFICE

#16
BT
1/2/02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANTS	:	Gary Mousseau; Mihal Lazaridis
TITLE	:	SYSTEM AND METHOD FOR PUSHING INFORMATION FROM A HOST SYSTEM TO A MOBILE DATA COMMUNICATION DEVICE
SERIAL NO.	:	09/782,380
FILING DATE	:	February 13, 2001
RELATED APPLICATION	:	Continuation of US 09/087,623 filed on May 29, 1998, now US 6,219,694
EXAMINER	:	Edelman, B.
GROUP ART UNIT	:	2153
ATTORNEY DOCKET NO.	:	555255012195

Affidavit of Prior Invention to Overcome Cited Patent Under 37 C.F.R. § 1.131

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

This Affidavit is submitted by the undersigned inventors to establish that the subject matter described and claimed in this application was invented prior to October 24, 1997, which is the earliest possible effective filing date of United States Patent Number 6,025,735 to Ulrich. The Ulrich patent was filed in the United States on April 10, 1998, but claims priority from two provisional applications, the earliest of which was filed on October 24, 1997. This Affidavit also established that the subject matter described and claimed in this

application was invented prior to November 18, 1997, which is the filing date of United States Patent Number 6,034,621 to Kaufman.

As this Affidavit demonstrates, the present invention was conceived of prior to October 24, 1997, and the inventors then diligently worked toward reducing the invention to practice and filing a patent application on this invention from prior to October 24,1997, to the effective filing date of this application, which is May 29, 1998.

We, Gary Mousseau and Mihal Lazaridis, the inventors of United States Patent Application S/N 09/528,495, titled "System and Method for Pushing Information from a Host System to Mobile Data Communication Device," and any related divisional or continuation applications, including United States Patent Application S/N 09/782,380 declare as follows:

1. Prior to October 24, 1997, we conceived of the invention described and claimed in this patent application. The following documents, which are attached to this Affidavit, were all created prior to October 24, 1997, and demonstrate conception of the claimed invention:

Tab A - Draft Patent Disclosure -- A draft patent disclosure entitled "A System and Method for Redirecting Mail from a Received Message Store" is set forth at Tab A. This Disclosure was created prior to October 24, 1997, and was subsequently submitted to our patent attorney, David B. Cochran, in December, 1997. The title of the invention was eventually changed to be "System and Method for Pushing Information from a Host System to a Mobile Data Communication Device." Note that Figures 1 and 2 of this early patent disclosure are almost identical to Figures 1 and 2 of the present application, and show real-time

information delivery between a primary message store and a secondary message store via a wireless network.

Tab B - Product Specification -- Prior to October 24, 1997, we had the document set forth at Tab B created by a third party. We met with this third party on several occasions in order to share our ideas regarding the invention. The third party then created this document using our ideas for implementing a system of real-time information delivery between a primary message store and a secondary message store via a wireless network.

Tabs C-D - Early Outreach Overview and Functional Specifications -- An internal technical document providing a high level overview of the "Outreach" redirector program, which is the subject matter of the document set forth at Tab B, is set forth at Tab C. Tab D sets forth a document entitled "Software Functional Specification." Pages 7 and 8 of this document describe message redirection features from a corporate server having a MAPI store to the mobile device. These documents, in part, formed the basis for generating the document set forth at Tab B. These documents were created well prior to October 24, 1997, and prior to the document at Tab B.

2. From prior to October 24, 1997 to May 29, 1998 (when the present patent application was filed) we diligently worked on reducing our invention to practice and also working with our patent attorney to prepare and file a patent application.

The documents set forth at Tabs E-K demonstrate additional work on the invention during this period of time. Tab E is an engineering specifications document. Tab F sets forth a series of design ideas for the Outreach product. Tab G is a software design

document for the Outreach product. And Tab H is another engineering specifications document.

Between October 24, 1997 and May 29, 1998, software development and coding work continued on the project, as evidenced by the documents set forth at Tabs I-K. Tab I is a printout from an internal software version control system for one of the software modules related to this invention. This printout shows that the initial version of the module was checked into the version control system on September 22, 1997 (page 11). Between September 22, 1997, and May 29, 1998 (when the present application was filed), more than 50 versions of the module were checked into the version control system. Likewise, Tabs J and K show that initial versions of the code were checked into the version control system around September 22, 1997, and that the code was continuously worked on from this time until May 29, 1998, and then for a substantial time thereafter.

In December, 1997, we provided technical documents and the draft patent disclosure, discussed above, to our patent attorney in order to prepare a patent application and subsequently thereafter, we discussed these documents and the invention with our attorney and in-house counsel. On January 5, 1998, our patent attorney completed a first draft of the patent application and sent this to us for review. The application was subsequently revised several times by us and our in-house counsel in conjunction with our patent attorney, and it was then filed on May 29, 1998.

3. We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so

made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

By: _____ Date: 31 AUG 01
Gary Mousseau

By: _____ Date: 30 AUG 01
Mihal Lazaridis

OutReach

Product Specification

Version 1.0

OCT 15 2001
PATENT & TRADEMARK OFFICE

Table of Contents

1. INTRODUCTION 3

2. FUNCTIONAL REQUIREMENTS 5

3. IMPLEMENTATION REQUIREMENTS 17

4. TEST PLAN 21

5. PROJECT PLAN 22

6. GLOSSARY 24

1. Introduction

RIM's Inter@ctive pager provides mobile customers with bi-directional communication capabilities, manifested as e-mail messages, a contacts database, and numerous calendar and scheduling features. Given that all of these capabilities exist within Microsoft Outlook, the inter@ctive pager's success will be larger determined by how tightly it is integrated with Outlook. OutReach is RIM's solution to this issue.

OutReach extends the user's desktop to include the inter@ctive pager. It gives the pager access to Outlook's data, extending the user's link to the outside world. This allows the pager to be a condensed, complete, and accurate reflection of the information managed by Outlook. This role has numerous implications.

Accuracy means synchronization must be maintained between the inter@ctive pager and the desktop. For example, if the contact information for a given client changes on either the desktop or the pager, the differences must be reconciled. This may be as simple as overwriting the obsolete record, or as complicated as blending the two records to accommodate overlapping modifications and race conditions. In any event, a solution to the synchronization problem is required.

Condensed means giving the user the ability to conveniently distill the potentially large volume of information managed by Outlook into useful elements, suitable for the mobility of the inter@ctive pager's notification paradigm. Such suitability is defined by hardware, bandwidth consumption, nature of use; in other words, defined by the pager's intended market

Complete means providing access to *all* of the information that is *useful* to the customer when he is away from his desk - e-mail, upcoming appointments, contact information, etc. The key to this requirement involves achieving a proper balance between feature content and practicality.

This document refines the extremely general requirements, mentioned above, into very specific requirements. The intent is to define the OutReach product at various levels and to various audiences.

The first level is the functional behavior of the product, often referred to as marketing requirements. Two questions of equal importance are answered: (1) What problems will OutReach solve and (2) what problems will OutReach not solve? This level is useful to everyone - sales, marketing, and engineering - because it "hangs the target."

The second level concerns how the functional behavior of the first level is to be achieved. This level enumerates implementation requirements, touching upon architecture, algorithms and the logical and physical structuring of the solution. This level is targeted for engineers and technical management. It not only conveys the details of the implementation but also the rationale behind the details. Ultimately, this section should

affect[1] everyone's concern for the product schedule by moving us closer to the problems and one proposed solution. An additional objective is to promote better risk management by demonstrating that the necessary thought process has occurred. It also calls attention to any technical uncertainties and vulnerabilities.

The third level is the mechanism used to declare the job complete - the test plan. The importance of a software development test plan cannot be overstated. It dictates the steps necessary to guarantee that the finished product is indeed finished.

The fourth and final level is the project plan. Here, the overall job is partitioned into more predictable tasks that are serialized in a schedule. This schedule contains milestones, deliverables and, ultimately, the target FCS date.

[1] Lower or raise

2. Functional Requirements

What problems will OutReach solve? What problems won't OutReach solve? The answers to these questions are crucial in that they define the product's *purpose* and *market disposition*. Purpose relates to the customer's perspective, while market disposition relates to RIM's perspective. All of the functional requirements for OutReach are discussed in the context of these two perspectives.

Another way to view these perspectives is to consider how they appear within the product. Requirements relating to purpose are close to the surface. Typically, the user interface reflects purpose through metaphors. Requirements relating to market disposition, on the other hand, tend to be more subtle and strategic. For instance, OutReach must be designed to support two modes of operation, client-based and server based. Both architectures solve the same set of problems to the pager; the differences surround our two perspectives. From the user's perspective, the product should not require the desktop to be online while the user is mobile. This calls for the server-based architecture. From RIM's perspective, the client-based solution targets the SOHO market while the server-based solution targets the corporate market. Moreover, the client-based solution can serve as an introduction into the corporate market. Each perspective has merit in that it places the product on a different plane.

The functional requirements for OutReach are presented independent of any implementation considerations. This is done to enforce the separation of *What* and *How*. There are advantages and disadvantages to this approach. The primary advantage is to prevent the two sets of requirements (functional and implementation) from overly restricting one and other. The functional requirements should only constrain the implementation regarding what is to be accomplished, not how it is to be accomplished. The primary disadvantage is the same; namely, separating the requirements may permit the functional set to be unrealistic. Since the advantages outweigh the disadvantages, the separation approach is used.

The most fundamental functional requirement of OutReach is that it supports only Outlook. All other requirements stem from this, which means the feature content of Outlook constrains the feature content of OutReach. This is a benefit because it provides an initial focus for our functional requirements. Consequently, a discussion of Outlook's features and concepts is appropriate, setting the stage for the presentation of OutReach's functional requirements.

2.1 Outlook Concepts and Features

Outlook organizes and presents its information using three basic concepts - *items*, *folders* and *views*.

2.1.1 Items

Outlook categorizes its data into one of six items. Each item is composed of numerous properties maintained within the Personal Folders Message Store. The six items are identified in Table 1.

ITEM	Contains this kind of information
Appointment	Appointments, meetings, and events; may be recurring or non-recurring
Contact	Names, street addresses, e-mail addresses, URLs, phone numbers, FAX numbers, and so forth
Journal	Log of phone calls, e-mail, and so on, with associated date and time information
Mail	E-mail messages
Note	Miscellaneous text
Task	To-do items, including information such as owner, due date, priority, and status

Table 1. Outlook Items

Because these items are maintained by the Microsoft's Personal Folders Store, OutReach may monitor and manipulate them through the standard MAPI interfaces. The implementation risks are minimized given that these interfaces are well defined and well understood.

2.1.1.1 Appointments

The rendition of an appointment is shown in Figures 1 and 2. Most of the significant properties of an appointment item are self-evident. An appointment has a subject, a location, a starting time, an ending time (i.e., duration), a reminder policy, and associated notes. An appointment may be categorized, for bookkeeping purposes, using either pre-defined or user-defined categories. The reminder policy dictates if, when and how a reminder is issued.

An appointment may have attendees that bind it to Address Book entries. When this feature is used, the appointment can be sent as an e-mail invitation. Subsequent replies are automatically processed upon receipt to facilitate scheduling and time management.

Perhaps the most important aspect of appointments concerns their scheduling implications. The Microsoft Exchange Server allows its users to publish personal schedules. This enables members of a work group to share this information to simplify the process of arranging meetings, conference calls, and the like.

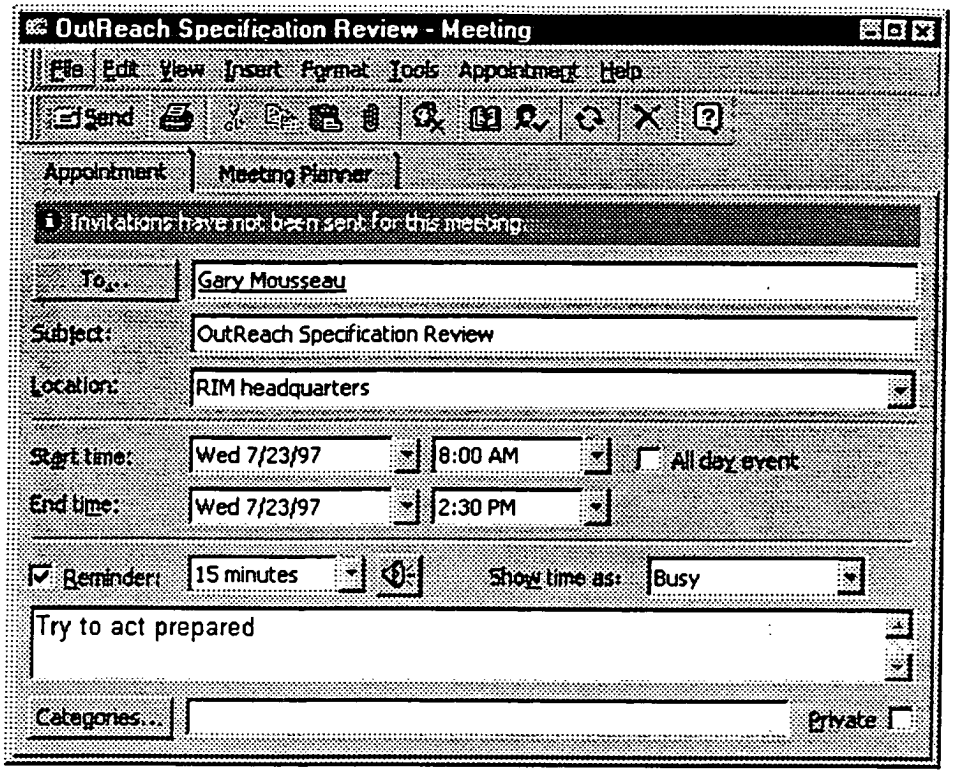


Figure 1. An Outlook Appointment

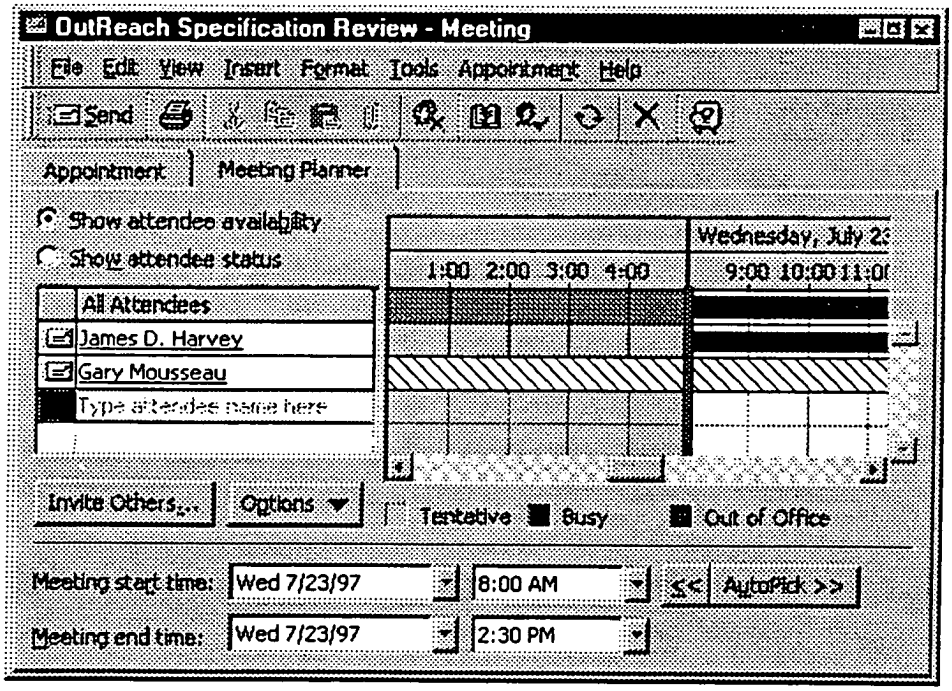


Figure 2. An Outlook Appointment (cont'd)

2.1.1.2 Contacts

Outlook maintains information about inter-personal relationships as contact items. Contact items not only record rolidex-like information; they also record interaction histories. These interactions are in the form of Journal items and have a many-to-one relationship to the contact.

An example of a contact is shown in Figure 3. The General property tab contains most of the information normally found on a rolidex card - name, company, title, postal address, phone numbers, e-mail address, etc. The Details property sheet contains information that is more obscure - spouse's name, birthday, profession, etc. The Journal property sheet lists interactions with the contact and can be filtered according to type. All phone call interactions can be viewed to the exclusion of all other interactions (meetings, conversations, etc.), for example.

The All Fields property sheet adds no additional information, but offers a flexible, low-level view of the contact. For OutReach, this provides a convenient means to define all of the data comprising a contact.

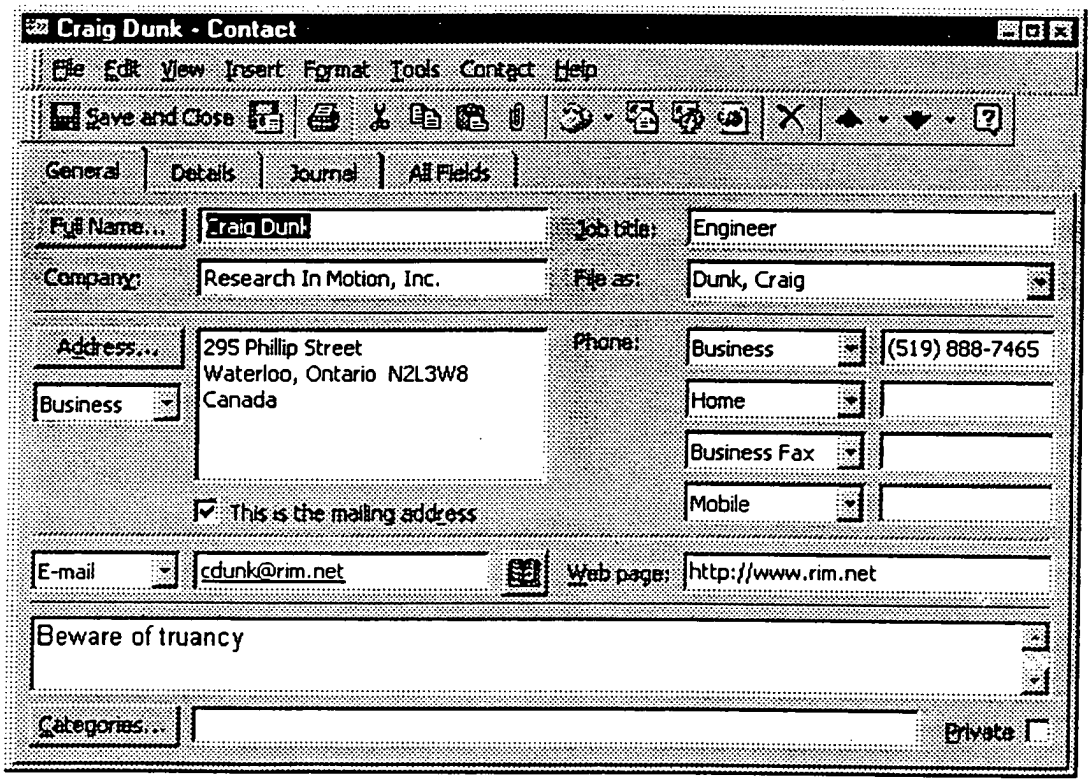


Figure 3. Contact.

2.1.1.3 Journal Entries

The Outlook user can record events such as phone calls, conversations, meetings, etc., into a journal. A Journal entry, shown in Figure 4, has a subject, a body, a contact, a category, a type and time information. The category helps the user easily find, sort, filter,

or group the journal entry in relation to other items. This allows the user to find *all* of the items related to a given project, for example; even when the items span multiple folders. In contrast, the type field is specific to the journal entry.

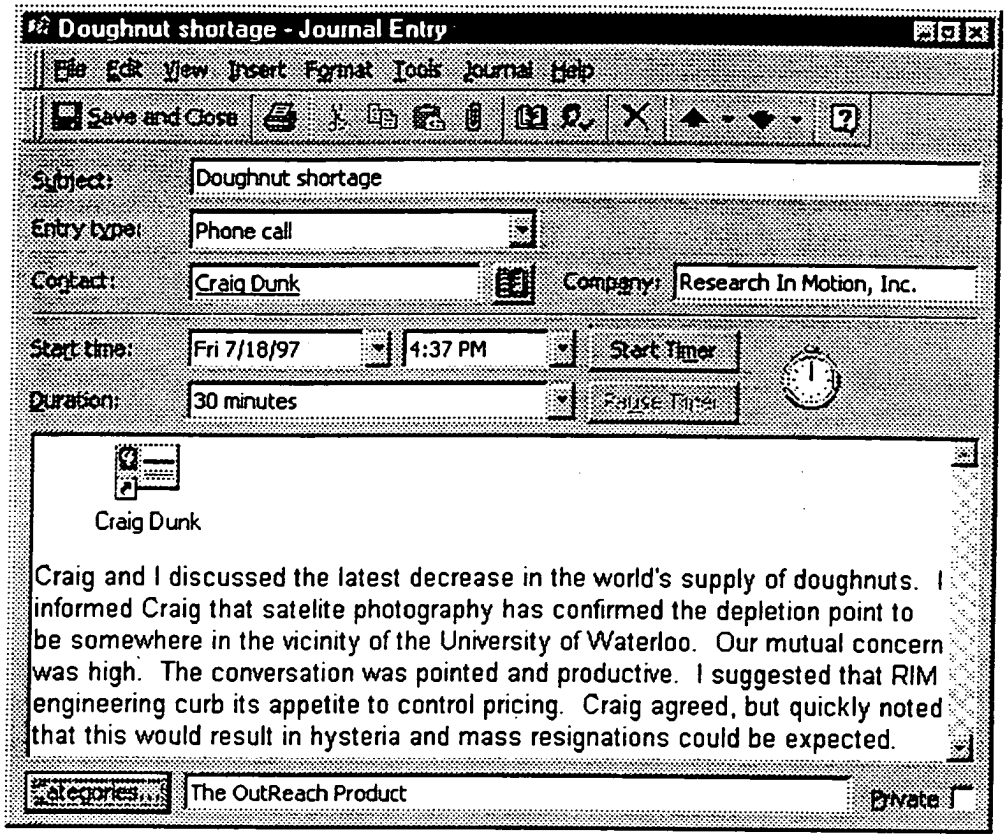


Figure 4. A Journal Entry

Journal items provide a very simple, intuitive means of managing time and organizing activities.

2.1.1.4 Mail

Messaging is a component of central importance to Outlook. It brings together all of the other types of items to form a tightly integrated, work group environment. As an example, Outlook users may assign tasks to one and other by simply mailing these items. Once a recipient *accepts* a task (via the message), status updates are automatically sent to the originator. This processing is integrated with Microsoft Project and Microsoft Team Manager.

The content and structure of messages in Outlook conform to the Extended MAPI structure. As such, their semantic capacity is broad and fully extendible within the Microsoft Exchange/Outlook environment. They can convey a subject, a body, 3 recipient types (To, cc, bcc), anonymous origination, 5 attachment forms including embedded OLE

objects, embedded messages, and much more.

In defining the functional requirements for OutReach, its best to focus upon what the interactive pager is capable of conveying and how Outlook e-mail messages interact with other items.

2.1.1.5 Notes

Miscellaneous text may be stored within Outlook as a note item. Notes provide a very simple, free-form method of recording unstructured data. They're rendered as a 3M post-it note. Although they may be forwarded to other users as e-mail message attachment, they add very little to Outlook's collaborative feature set.

2.1.1.6 Tasks

Tasks are an important element of Outlook's time management features. They formalize things to be accomplished. Tasks are well defined, structured records containing a subject, a body, an owner, a current status, a priority, a completion state, a reminder policy, a category, and (optionally) a due date. A task example is illustrated in Figure 5.

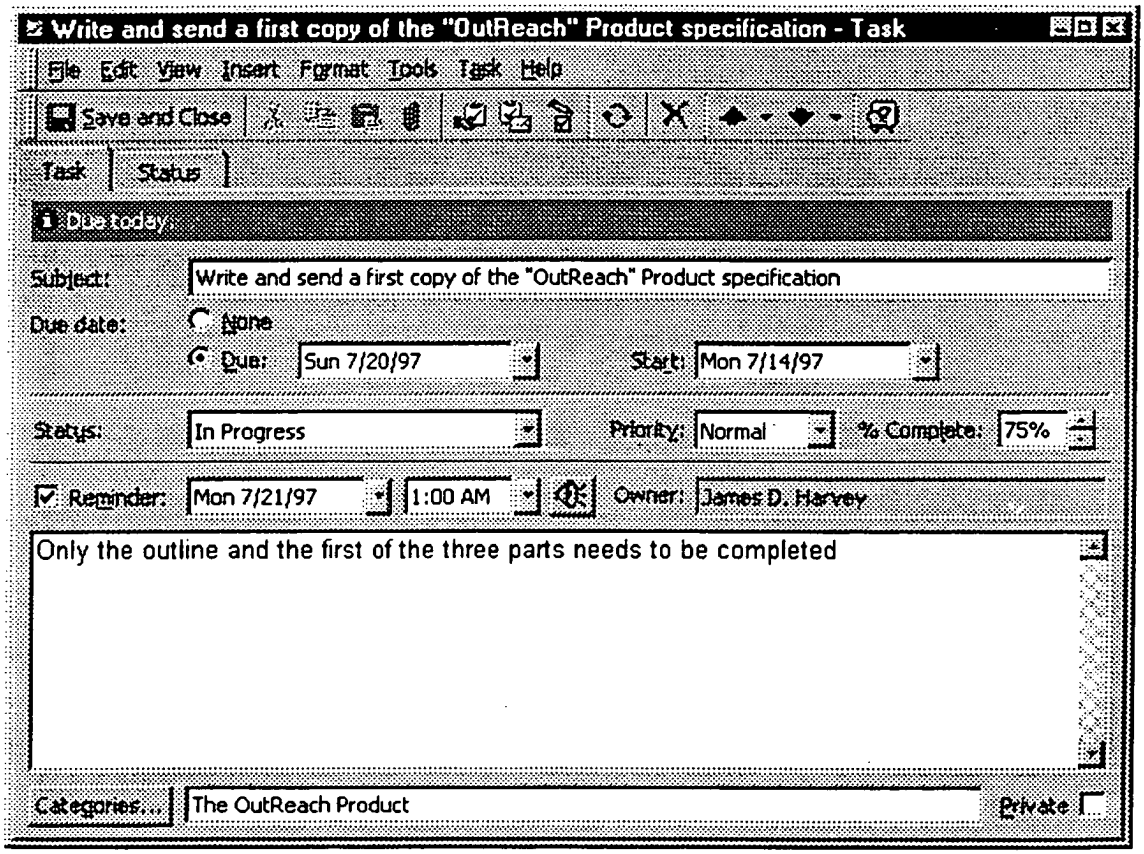


Figure 5. An Outlook Task

Tasks may also be used in a work group manner when combined with messaging. An Outlook user may assign a task to a colleague by designating that colleague as the owner and sending the task to that same person. This capability requires either Microsoft Project or Microsoft Team Manager.

2.1.2 Folders

Items are stored according to their type within the Message Store using Folders. Eight, pre-defined folders (listed in Table 2) ship with Outlook to support all six items types.

This Folder	Contains this item type
Calendar	**Appointment**
Contacts	**Contact**
Inbox	**Mail**
Journal	**Journal**
Notes	**Note**
Outbox	**Mail**
Sent Items	**Mail**
Tasks	**Task**

Table 2. Designated Outlook Folders

2.1.3 Views

Collections of items are presented to the user through views. In MAPI, a view is bound to a folder through the folder's Associated Contents Table. For the Outlook folders, there are pre-defined views and user-defined views. The only significance of views to OutReach concerns look-and-feel. Obviously, the more the interactive pager's user interface resembles Outlook's views, the more familiar the user will be with the pager when first learning to use it.

To the Exchange Client, the concept of a folder view is more rudimentary than it is in Outlook. The view is defined according to the columns shown with the MFC ListView Control. In Outlook, the definition of a view appears to be more complex. For example, the pre-defined view for journal items is shown in Figure 6. The entries are presented in a time-line manner and organized according to type.

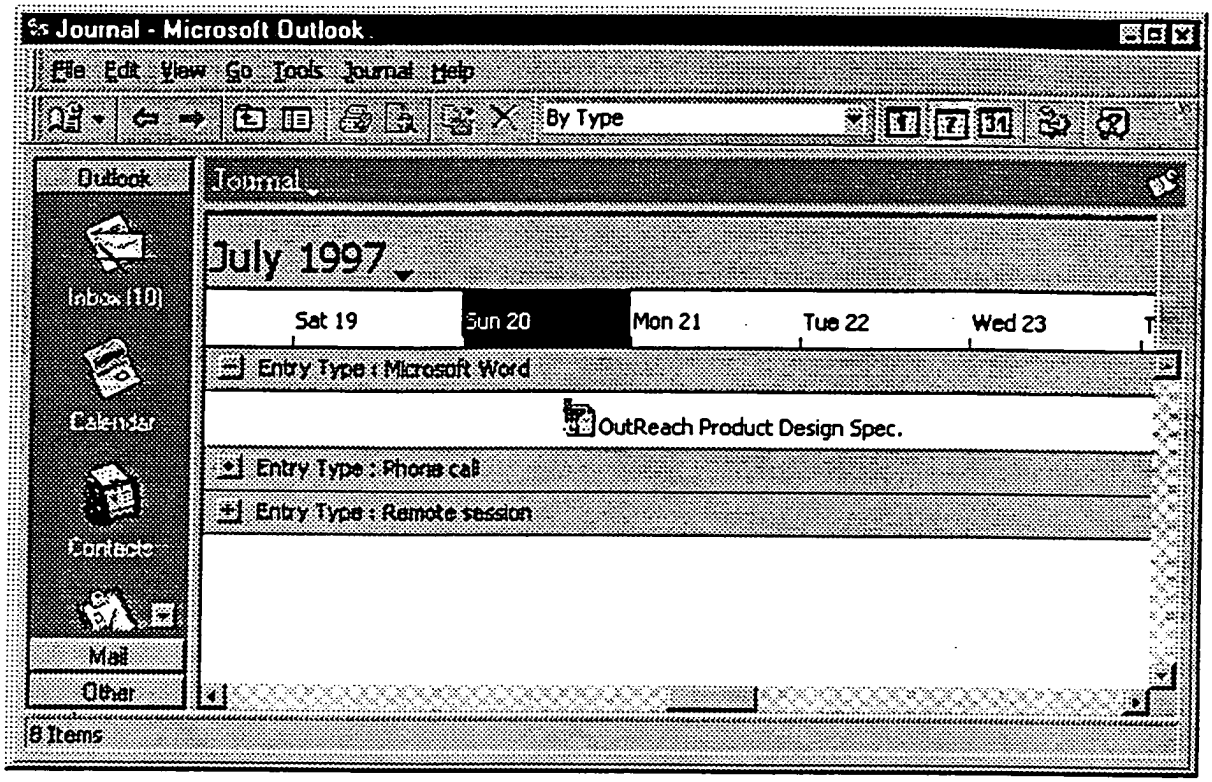


Figure 6. The Default Journal View

2.2 The User's Perspective

With all of the above Outlook concepts and features in mind, the following functional requirements are placed upon OutReach. All of these requirements reflect the user's perspective. In other words, what problems will OutReach solve for the customer?

OutReach shall possess the following features:

2.2.1 Out-of-Box

1. The OutReach out-of-box experience shall be simple and clear. The SETUP program shall guide the user through the installation process using a wizard metaphor. Specifically, the program will ask the user a series a very simple questions that require only very simple answer's. The answers to these questions shall lead the user to a functioning state. The intent is to install the product on the user's system while shielding the user from as much complexity as possible.

2. The SETUP program shall give the user the option to perform the installation with either a serial or wireless connection to the Inter@ctive pager. Serial will be the default since this will not require the user to know the MAN number of his specific pager.

3. The Inter@ctive pager's MAN number will be detected automatically when the serial connection SETUP mode is used.

4. The SETUP program will allow the user to populate the Inter@ctive Pager's Address Book. This may be accomplished in a bulk manner - where all Outllok contacts and Address Book entries are downloaded to the pager - or on an entry-by-entry basis.

5. The SETUP program shall allow the user to download his Outlook Calendar to the Inter@ctive pager.

6. The SETUP program shall give the user the option of customizing the Inter@ctive Pager's Phrase Book.

2.2.2 E-Mail

7. OutReach shall relay all Interpersonal[2], Outlook, e-mail items between the inter@ctive pager and Outlook/Exchange.

8. The originator's identity for e-mail messages sent from the inter@ctive page will be identical to those sent from the user's desktop. More specifically, the customer's use of the inter@ctive pager shall be transparent to the outside world with regard to outbound messages.

9. OutReach shall accept read notifications from the inter@ctive pager. These transmissions indicate that a specific message was read on the interactive pager. In response, OutReach shall mark the corresponding message as read on either the desktop or server. The intent is for OutReach to support fully the read bit of the PR_MESSAGE_FLAGS message property. Note that many third party Exchange applications rely upon the proper usage of this property. Note that this will be a per-message option on the Inter@ctive pager. In other words, read notifications will

10. OutReach shall only support e-mail message content that may be rendered as ASCII text. This means that rich-text-format will not be supported, nor will binary attachments. This also pertains to all of the other MAPI properties that comprise the message.

11. OutReach shall support two methods of messaging forwarding - Complete (subject to filtering) and Summary (header only). The use of either method shall be subject to requirement 16 (see section 2.2.6).

12. OutReach shall support message augmentation. Specifically, OutReach will add the original message body to all replies/forwards coming from the pager. This will be configurable option.

---

[2] those having a PR_MESSAGE_CLASS property value beginning with "IPM."

2.2.3 Address Books and Contacts

13. OutReach shall allow the customer to download contact items and any other Address Book entries from either the Outlook Desktop or Exchange Server to the inter@ctive pager.

14. OutReach shall remote Name Resolution (e.g.., search for Gar*). This feature shall allow the user to (effectively) request Address Book updates from the pager

15. OutReach shall allow the user to monitor selected contact items or entire contact item folders. This monitoring will detect modifications to these items and will relay changes to the inter@ctive pager.

16. OutReach shall accept contact item modifications from the inter@ctive pager and shall update the corresponding contact items on either the Outlook Desktop or the Exchange Server.

2.2.4 Calendar and Scheduling

17. OutReach shall support appointment items. This means that the appointment invitations and responses shall be supported in both directions.

18. OutReach shall support the free/busy time, bi-directional, scheduling interrogations. This means that OutReach will permit the inter@ctive pager to make free/busy time queries any published schedule. It also means that OutReach will allow the inter@ctive pager user to publish his schedule, as maintained on the pager, to others.

2.2.5 Journal

19. OutReach shall, as a configurable option, translate inbound inter@ctive pager communications having a specified category into journal items. This will be in addition to the processing normally done according to the nature of the communication. For example, when OutReach receives an e-mail message from the interactive pager that has a category, the receipt of the message will be recorded as a journal item before the message is relayed to its ultimate destination. The intent is to allow the user to augment his desktop journal from the pager.

20. OutReach shall allow the user to create journal entries to reflect his activities in the field. This feature will allow the user to not only define an activity, but also designate its duration. The duration will be derived from remote commands that (a) start the activity, (b) suspend the activity, (c) resume the activity and (d) complete the activity. Each of the aforementioned commands will convey time.

2.2.6 Tasks

21. OutReach shall support the communication of task items between the inter@ctive pager and Outlook/Exchange. This implies (among other things) that user's of

Microsoft Project and Microsoft Team Manager may assign tasks to colleagues through the inter@ctive pager. The inter@ctive pager user may either accept or reject the task item, which OutReach will convey to the originator accordingly. This also means that OutReach shall support Task Status updates between the pager and Outlook.

2.2.7 Configuration

22. OutReach shall allow the user to control bandwidth consumption. This control will be in the form of item size restrictions, item age restrictions and item originator filtering.

23. OutReach shall support encryption and compression for communicating with the inter@ctive pager.

24. OutReach shall maintain a list of preferred originator addresses in the form of a workgroup. Messages from workgroup members shall be forwarded to the Inter@ctive pager by default. In contrast, only the headers of messages sent from outside the user's workgroup will be forwarded to the pager.

25. OutReach shall allow the user to edit the InterActive Pager's Address Book. Serial and wireless connections to the pager may be used.

26. OutReach shall allow the user to edit the InterActive Pager's Phrase Book. Serial and wireless connections to the pager may be used.

27. OutReach shall allow the user to edit the InterActive Pager's Calendar. Serial and wireless connections to the pager may be used.

28. The user shall be able to modify the OutReach configuration from the Inter@ctive pager. At a minimum, this capability shall support:

    (bb)Enabling/Disabling the forwarding of all messages;

    (cc)Modifying the message size threshold; and

    (dd)Modifying the workgroup membership.

2.2.8 Miscellaneous

29. OutReach shall allow the customer to download item categories -- both user-defined and pre-defined -- to the inter@ctive pager. This will permit the pager user to attribute a category for each transmission.

30. OutReach shall allow the user to monitor individual Public Folders on the Exchange Server and individual items contained within those Public Folders. This monitoring will detect changes and will communicate these to the inter@ctive pager. For Public Folders, new items will be detected and forwarded to the pager. When individual

items are modified, the items will be relayed to the pager.

31. OutReach shall (optionally) provide a screen saver UI that allows other people to conveniently communicate the user from his own desktop. This requirement serves two purposes (1) to allow colleagues to readily recognize that the user is accessible via the inter@ctive pager and (2) to promote the easiest way to (informally) communicate with the user when he is mobile.

32. OutReach shall provide warnings when a request is anticipated to yield abnormally large pager transactions. For example, if the user requests to download the entire Global Address List from his Exchange Server (having many entires), a warning will be sent from OutReach to the pager, asking the user to confirm the operation.

2.3 RIM's Perspective

The following functional requirements are placed upon OutReach and reflect RIM's perspective

OutReach shall possess the following features:

1. OutReach shall support a dual architecture - a desktop-based mode and a server based mode.

2. In desktop mode, OutReach shall require an active MAPI profile.

3. Any rendering of items through the desktop UI shall be subject to configurable security restrictions. In other words, the desktop UI shall only provide visual access to Outlook items if configured to do so. This mean that while the customer is using the inter@ctive pager (away from his desk), he can control how much data is visible through his desktop.

4. In server mode, OutReach shall not require the user's desktop to be online.

3. Implementation Requirements

3.1 Look-and-Feel

The two versions of OutReach - desktop and server - will have a similar, Explorer-like user interface. A TreeView/ListView Splitter window will be used. The tree in the left pane will segregate the Outlook items according to type. The list in the right pane will enumerate messages processed by OutReach. The root of the tree for the desktop version will contain a single glyph representing the Inter@ctive pager. In the server's UI, the root of the tree will contain a glyph representing all of the active pager's in the field. The immediate children of the root will be identical to the desktop.

An example of the desktop UI is shown in Figure 7.

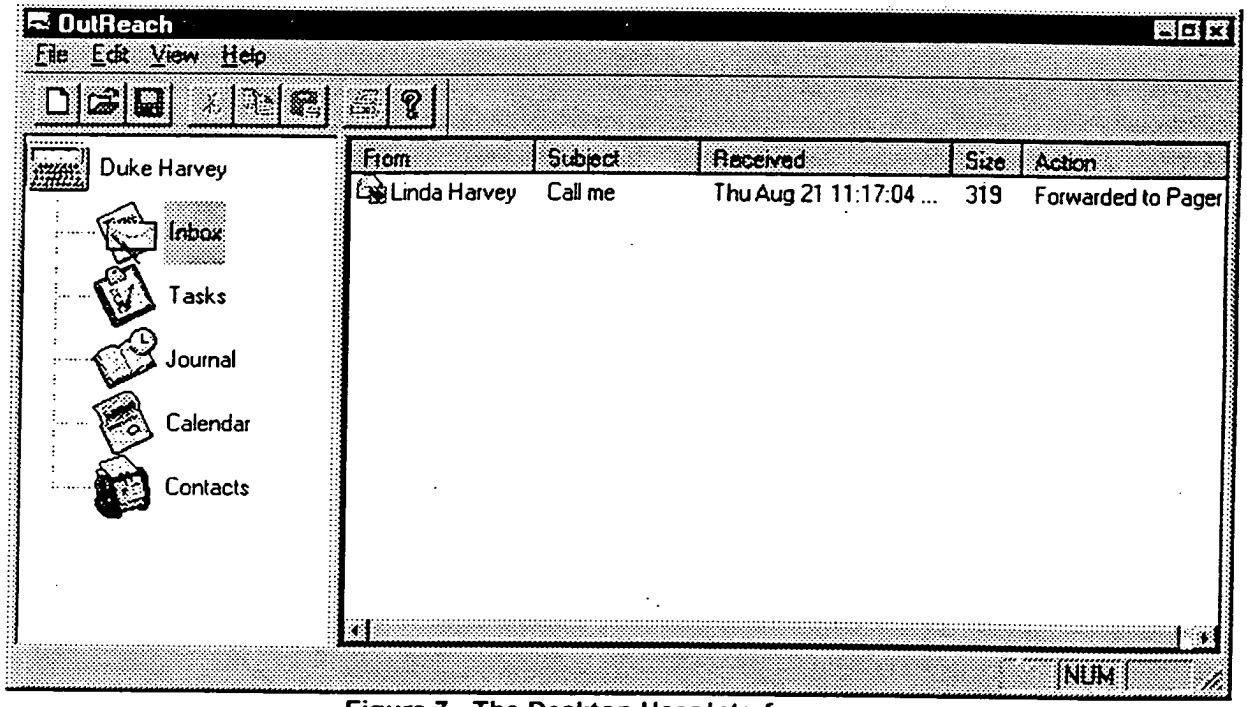


Figure 7. The Desktop User Interface

The icons used in the right pane reflect the described action. In the above example, the message "Call me" was forwarded to the pager, so it's icon is an envelope with a blue arrow. Except for the color of the arrow, this is the same glyph used by Outlook to designate a forwarded message. Other forms of icons are anticipated. The icon used for a message not forwarded, due to filtering, will be a plain envelope (without an arrow). For a message that has been truncated, due to size restrictions, the icon will be a "torn" envelope with a forward arrow.

OutReach's paradigm will be that of a communications log; effecting, recording and rendering all communications to and from the Inter@ctive pager. As such, the user may

review these communications upon completing an Inter@ctive session. To this end, OutReach will support message rendering, but this will be subject to configurable security restrictions. Such restrictions will form a security gradient. Under the least secure setting, items will be rendered when double-clicked within the right pane. This rendering will use the same OLE Form servers used by Outlook. The rendering of the "Call me" message in Figure 7, is shown in Figure 8.

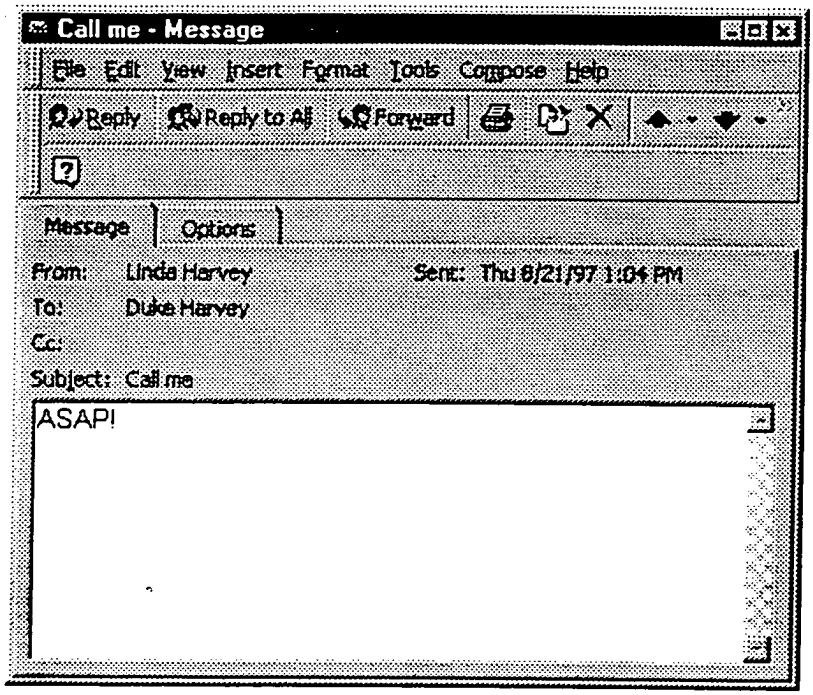


Figure 8. Rendering an IPM.Note.

The next level of security will allow password-based rendering. In this scenario, when the user double-clicks a message within the right pane, he will be asked to provide the correct password. This safeguards the user's messages from unauthorized inspection.[3]

The highest level of security will encompass the previous level and will also allow the user to restrict the columns visible within the right pane. This would allow the user to hide sender identities and subject lines.

OutReach will support numerous filtering features to permit the user to control his wireless bandwidth consumption. A maximum message threshold will protect the user from receiving overly large messages. OutReach will also support message header forwarding coupled with the ability to define workgroups. When the user adds a person

[3] For consideration: If the user fails to enter the correct password, OutReach could send a "silent" notification to the Inter@ctive pager - notifying the user of the attempted security breach. This could be a configurable option - password protected ofcourse.

to his workgroup, all messages from that person will be forwarded to the pager, automatically. For all other senders, only headers will be forwarded to the pager. The residual data of these headers may be subsequently downloaded to the pager upon request.

The OutReach server will offer a special feature to enable the user to review all of the communications of an Inter@ctive session. This capability will work as follows:

1. When the user returns to his office at the conclusion of an Inter@ctive session, he communicates this to the OutReach server.[4] The purpose of this communication is to instruct the server to discontinue it's forwarding service.

2. Upon receiving this directive, the OutReach Server will copy the portion of the communication log pertaining to the user, and will send this to the user as a message attachment. The file extension of this attachment will be the registered type of the OutReach OLE Server.

3. When the user receives this message, confirming the end of his Inter@ctive session, the attachment will appear as the OutReach program icon. By double-clicking this icon, the user may review all of the messages exchanged during the session; this would include messages truncated or not forwarded due to filtering.

This feature would provide a very convenient means for the user to review messaging events that occurred while he was away.

3.2 Architecture

OutReach's user interface will employ the standard MFC Document/View Architecture. The Document will be an OLE Compound Document. For the desktop version, there will be a one embedded IStorage object containing all of the communications to and from the user's pager. For the server version, there will be one embedded IStorage object for every active pager in the field.

Most of the UI code will be re-usable between the two versions of the product. Furthermore, all of the code surrounding the OLE Compound File (i.e., the communications log) should be applicable to both implementations. Beyond these points, however, the two implementations diverge.

The most significant architectural issue of the server concerns it's message access. The server is a bonafide e-mail gateway in that it relays messages and translates content between two disparate representations. Accordingly, message access could be achieved by implementing the server as a true gateway having its own address type. A second alternative is to implement the server in a manner similar to the RAD-Exchange server. A

---

[4] Using an Outlook client extension.

main thread would receive e-mail messages from users to start and stop the forwarding servuce. This thread could spawn a *mailbox* thread when activating itself for a specific recipient.

3.3 Key Algorithms

Software algorithms for key creation, manipluation and usage will be provided by RIM. Key management, especially in the server model, will be handed by @ware and the Outreach product.

RIM will also provide routines for compression and encryption of data being send to, and being received from, the Leapfrog.

4. Test Plan

4.1 Profiles

4.1.1 Feature

RIM and @ware will both be creating a list of features that will be used for creating a test plan. This test plan will include a simple regression test of all features and the expected results from each test.

4.1.2 Stress

RIM will create a model for stress testing the Outreach product. This will be integrated to the feature-based testing stage. The stress test will test both the desktop and server products.

4.1.3 Negative

@ware will be primarily responsible for creating tests to check 'negative' responses to incorrect user input. RIM will be responsible for creating coverage and other conditions.

4.2 Coverage Matrix

A test plan has little value if it does not include a coverage matrix. A coverage matrix correlates test cases to requirements. The purpose is to guarantee some degree of completeness.

The product requirements appear on one axis and the test cases appear on the other. A check mark is placed in each matrix cell if the corresponding test case addresses the corresponding requirement. If each requirement row[5] contains at least one check mark, then some degree of completeness has been achieved.

[5] assuming the requirements are on the y-axis

5. Project Plan

The purpose of any project plan is to partition a large, less manageable problem into smaller, more manageable sub-problems. The intent is to decrease scheduling risks by isolating tasks, complexities and costs. The project plan for OutReach is presented in this spirit.

The partitioning process begins with the definition of feature sets and product profiles. A feature set defines a collection of features that are related logically. Every feature set has a cost that is calibrated in time and money. Feature sets are combined to create a product profile. Each product profile defines a certain level of product sophistication.

Table 1 enumerates all of the feature sets for OutReach.

Feature Set	Description	Cost	Risk
Message Relay	The ability to relay normal (IPM.Note) messages between OutReach and the Inter@ctive Pager. No filtering or header capabilities are provided	$20,000	Minimal
Filtering	The ability to control when and how messages are exchange between the OutReach and the pager. This includes a message size threshold, message header transmissions, and workgroup definitions.	$5,000	Minimal
Security	The ability to control how much information is revealed through the user interface. This involves password protecting message access, column viewing and configuration. Security attacks are also detected and reported.	$5,000	Minimal. **NOTE: Save for Stage 2**
Scheduling	The ability to request, reject and accept meeting requests.	$5,000	Moderate
Addressing	The ability to download Address Book entries and Contacts from Outlook to the Inter@ctive Pager.	$5,000	Minimal

Journal	The ability to define journal entries from the pager and to record the duration of these remote activities. This includes the ability to start an activity, suspend an activity, resume an activity and complete an activity.	$5,000	Minimal
Tasks	The ability to assign, accept and reject tasks and to inter-operate with Microsoft Project and Microsoft Team Manager.	$5,000	Moderate
Synchronization	The ability to detect changes to Public Folders as well as changes to the Global Address Book. Also included is the ability to update Public Folders, the Global Address Book, and the Personal Address Book from the field. On-demand reconciliation of information including Calendar and Personal Address Book. This shall be based upon the maintenance of a time-stamp by OutReach and dirty-bit(s) by pager.	$5,000	Moderate
OLE Server	Implementing the OutReach Desktop as an OLE Server and supporting session recall from the server. Session recall is when the server sends the communication log, as an attachment, to the user at the end of an Inter@ctive session. These documents could be used for trip reports, billing and accounting, etc.	$20,000	Moderate **NOTE : Will save for Stage two!**
Pager Configuration	The ability to configure the pager from OutReach. This would provide an alternative to doing this through the pager itself. This will be an Outlook client extension in the server model and a normal UI command within the desktop model. On the desktop, this may involve either serial or wireless communication with the pager.	$7,000	Moderate

Setup	Automated installation of the product, including serial and wireless communication with the pager.	$7,000	Moderate
Server Framework	The ability to run OutReach as a agent on the Exchange Server..	$11,000	Moderate
Bells & Whistles	Implementing a screen saver to allow pages to be easily sent to the user while he is Inter@ctive..	$5,000	Moderate
TOTALS	Removing the Stage two Developments	$80,000	

Table 3. OutReach Feature Sets

5.1 Milestones

The following is a brief summary of the deliverables, milestones and payment schedule.

Milestone	Date	Payment	Description
Prototype	12-Sept-97	20%	Proof of concept and Contract Signing
Alpha1	31-Oct-9	15%	A functioning version that includes Message Relay, Scheduling, Addressing and Filtering
Alpha2	28-Nov-97	15%	Alpha1 plus Pager Configuration, Tasks, Journal and Synchronization.
Beta1	19-Dec-97	15%	Alpha 2 plus a functional Setup program and a functional server model.
Beta2	16-Jan-98	15%	Beta1 plus fit and finish issues resulting from Beta1 feedback.
FCS	30-Jan-98	20%	Production-quality, ready for general distribution

Table 4. Milestones

5.2 Product Profiles

The features enumerated in Table 3 may be combined in various ways. Each combination yields a product profile. This profile defines functionality and cost.

6. Glossary

Bug - (1) *Engineering noun* - An emerging feature. (2) *Marketing and Sales noun* - Software. (3) *Obfuscatory noun* - See Feature

FCS - First Customer Shipment; when a production-quality version of the product is available for distribution to the general public.

Feature - An Engineering term (refer to Bug).

Item - The fundamental building block for the information maintained by Outlook. There are six types - appointments, contacts, journal entries, e-mail, notes and tasks.

SOHO - Small Office / Home Office

Introduction

The main purpose of this document is to act as a foundation for a development contract for creating the 'Mail Redirector' product

Technical Overview

The I@P mail redirector is a Microsoft MAPI extension built on the main MAPI clients, Exchange and Outlook. As a result of using the redirector a user will be able to set up new routing information to get mail, address book and schedule information to their I@P product. To help eliminate the necessity of keeping the user's computer on 24 hours a day, 7 days a week, there will also be a server version created as well. The server version will keep a full cache of all mobiles and allow for one administrator to setup the entire system. The server version will be considered phase 2 of the project, while extended schedule and computer control will be considered phase 3.

Functional Summary

The Redirector will be made up of several important components these include:

1. A User Interface component, which is primarily for allowing the user to edit configuration parameters and for letting the user selected.
2. A directory synchronization component, that can display the current address book and allow the user to select which addresses get sent to the pager.
3. An E-Mail component, that detects new MAPI messages

Software Functional Specification

Version 2.1

Product : Leapfrog
By : Gary Mousseau

Proprietary and Confidential

Table Of Contents

Introduction 3

1.0 Common Paging Features 3

Message Reception 3
Message Transmission 4
Message Viewing - Summary List 4
Message Viewing - Detailed View 4
Address Book - General 4
Phrase Book (Canned Messages) 5
Import Facility 5
Application Options 5

2.0 Advanced Pager Features 6

Message Exchange - Reception 6
Message Security 6
Message Compression 6
Message Origin and Transport Options 7
Messages from POP3 Connection 7
Messages from a MAPI Store 7
Alternative Pager Applications 8
Real-time Browsing Support 8
Schedule Tracker 9
Import Facility 10
Current Pager Enhancements 11
Application Behaviour 11
Message Exchange - Viewing 12
Address Book - General 13
Phrase Book 13
Expanding The Pager's API 14
NBS Changes 14
New UI Engine 14
Complete Application API 15
Adding Racotek's Key-Script Language 15

Appendix A : Wireless Host Access 16

Internet Access Phase 2 16
Internet Access Details 16
Internet Access Changes Required 16
Alternative Connections 17

Introduction

The following is a terse summary of the software options within the Leapfrog product, what is being done and what could be done with the product. Since RIM no longer wants to create the majority of software within the Leapfrog product, these items could also been seen as software that RIM would desire other partners to write, as part of their end-to-end service offering. This document will describe and highlight the meaning and market justification for basic and advanced software features within the Leapfrog. As a result of this document a set of features will be identified and targeted for the product and then promoted to our software partners. One of the biggest decisions that must be made early on is which wireless network will be selected for the first release of the product, Mobitex, DataTAC or CDPD. Naturally with no confirmed orders or contracts for two-way pagers in CDPD, a decision to build for those networks first entails the highest risk.

Another **big** decision that RIM has made recently is to include a simple transport layer within the Pager's API. This simple transport layer will allow more than one packet to be delivered reliably to the device. The transport chosen was based on the Narrowband Sockets (NBS) specification from Intel and Nokia. This will allow a full-featured client to be built over both Mobitex and DataTAC by tools manufacturers like NetTech and RacoTech. These companies will in turn bring clients like Wynd to the product through their developer relationships.

1.0 Common Paging Features

There are some common features that are part of Leapfrog across different network types. These common features will be examined first, followed by the impact each network will have on how the basic features operate. Therefore this section will examine each network in turn and examine how each feature will behave. Even if a small difference is present the feature will be included in this section, and a reference will be made highlighting the difference between the networks.

Message Reception

FEATURE: Messages that are received are always interpreted as mail messages, based on which protocol is being used. Even messages that originated as Faxes, or Voice-to-Text messages are received and treated as basic mail messages. When a mail message is detected, it is stored into the flash memory for permanent storage. Additional features include:

a) Currently a message that is received can contain address book information that can be imported into the address book.

b) Currently a message can contain configuration information that can change user profiles on the pager.

BENEFIT: Messages are securely stored and available even after changing batteries and in most cases even across Pager resets

STATUS: This feature, in its simplest form, is already part of the current Inter@ctive Pager product. Therefore the feature will be ported when we move the current application to the Leapfrog.

LIMITS: For RAM, using the HP-ID 4 protocol and a fixed maximum packet length of 512 bytes. For Ardis using Ardis Personal Messaging (APM) the maximum message size if 2048 bytes. There is no compression for cost savings, and no encryption for security being used.

See the advanced section for additional features and possibilities with the RAMFirst service.

Message Transmission

FEATURE: Messages that are sent can be transmitted to the gateway using different techniques or message indicators. These message indicators include the following types: Inter@ctive Pager - for peer-to-peer mail or Paging, Fax - for Facsimile transmission within the gateway, Voice - for translation into synthesized voice within the gateway and one-way - for sending the message to a one-way pager.

a) The user also has the ability to send address book entries as mail messages. This allows other users to be synchronize address books between devices.

b) Message can be composed and placed in a 'saved folder' or send immediately. The Pager will watch for coverage to be present before it tries to transmit the message.

BENEFIT: The user is given the feeling of having more flexibility and power by allowing them to send a range of message types. The end result is that many services are combined into one device.

STATUS: This feature, in its simplest form, is already part of the current Inter@ctive Pager product. Therefore the feature will be ported when we move the current application to the Leapfrog.

LIMITS: For RAM, using the HP-ID 4 protocol and a fixed maximum packet length of 512 bytes. For Ardis using Ardis Personal Messaging (APM) the maximum message size if 2048 bytes. There is no compression for cost savings, and no encryption for security being used. Currently a message can only have one destination address, this means I have to resend the message over and over to many destinations.

Message Viewing - Summary List

FEATURE: Once a message is stored in the flash memory it can be viewed. By default all messages 'Sent' or 'Received' are viewed as one chronological list of messages and icons are used to distinguish whether they are part of the Inbox (Received) or Outbox (sent) messages. It is possible to change the view and select only Inbox or Outbox messages.

BENEFIT: Users can quickly locate their latest messages, the default 'enter' key action also allows browsing new messages easier as well.

STATUS: The basic 'all messages' view, 'Inbox-Only' view and 'Outbox-Only' view are complete and working on the Inter@ctive Pager today.

LIMITS: When the view is changed to Outbox it is impossible to view new incoming messages.

Message Viewing - Detailed View

FEATURE: During the viewing of the summary list of messages the user can select one and open it for detailed examination. In this mode the user can scan the message, reply to the message or forward the message.

BENEFIT: This feature allows the user to easily scroll through and view the entire message. It also allows the user to reply, forward or delete the message while it is viewed without returning to the summary list.

STATUS: The basic detailed view functionality is complete on the Inter@ctive Pager and will be ported to the Leapfrog.

LIMITS: Messages are only 512 bytes long for Mobitex and 2048 bytes long for DataTAC. There is normally no subject field and 'to' field shown for the message.

Address Book - General

FEATURE: The address book is one of the most common elements across the networks. The goal of the address book is to hold E-Mail addresses and contact information for the user. For RAM it is possible to use the addresses to hold routing information as well. For example an address that has a format "Number@FAX" format indicates to certain gateways that the address is a fax number and the information is to be faxed out. It is also possible to 'send' address book entries as mail message to another recipient, which could be another pager or a gateway.

BENEFIT: The address book allows the user to save time in constructing the destination address since they doesn't need to remember address, phone and other contact information. Additional contact information can be stored on the pager, thus making it a simple contact manager as well. Being able to send and receive address book updates makes getting addresses onto the device much easier.

STATUS: This feature is currently available in the Inter@ctive Pager product, and a basic address book will be ported to the Leapfrog.

LIMITS: Extra work is required to supporting taking multiple address book entries and placing them onto an outgoing message. There are no folders simply one large address list. There is a direct update method for updating the address book but the configuration tool must be used for this. The import/export capability is an excellent start to a full synchronization functionality.

Phrase Book (Canned Messages)

FEATURE: The phrase book is used to hold standard phrases or auto-response messages. Phrases can be automatically loaded, changed or added to by the user.

BENEFIT: The end user can quickly reply to messages in situations where the keyboard would be difficult/impossible to use (i.e., driving). This feature is available across all Mobitex server options, it is not likely to change based on the back-end server that is selected.

STATUS: This feature is currently available in the Inter@ctive Pager product and will be available in the Leapfrog.

LIMITS: Having different folders for canned messages would make it easier to have larger amounts of canned messages.

Import Facility

FEATURE: The import facility allows the user to receive standard mail messages and import their contents into the address book or into the phrase book. The format and contents of the message currently are fixed and the sender of the message must know this format.

BENEFIT: The import would allow a user to quickly and easily create a complete address/contact book and phrase book when buying a new pager. Improving the import facility would allow for more flexibility in the variety of formats and data types that could be received.

STATUS: A limited import feature is currently available in the Inter@ctive Pager product and will be ported to the leapfrog.

LIMITS: There is no way to automatically import into the Address Book or Phrase Book, currently it is a manual step. The lack of a schedule package leaves a major hole in the product, but if it were present then importing into scheduling software would be critical. Naturally the whole issue of the back end must be resolved, i.e. tying the synchronization to a widely used software product like Schedule+, Outlook or Goldmine.

Application Options

FEATURE: This feature allows the user to customize the application and tailor certain behaviour for regular operations. This customization is in the current Inter@ctive Pager and is not expected to change dramatically.

BENEFIT: Adding additional options allows the user to customize the product further to their needs.

STATUS: A basic form of this feature is currently available in the Inter@ctive Pager product and will be ported to the Leapfrog.

LIMITS: Expanding these options to add further customization would be important. Depending on the features added by third parties these options may have to be extended to allow other programs to 'hook' features into the list.

2.0 Advanced Pager Features

This section describes the possible advanced features that could be implemented based on interested third party developers and upon the server that is selected.

Message Exchange - Reception

Within the area of message reception there are the widest possible advanced features possible. Naturally as third parties start development these areas will begin to be developed and expanded.

Message Security

FEATURE: Currently there is no security in the Inter@ctive Pager product. This includes basic device authentication or message security for incoming messages.

a) One option is to add the encryption protocol being used in the RAMFirst gateway, this is a shareware product call enigma. This encryption method can be used on top of the protocol that is current being used, that being HP-ID 4.

b) Design another encryption method to the pager and those that want to use it can license the technology. A current favourite choice is Elliptic Curves currently being licensed by Certicom. In this way the Pager 'sets the standard' or encryption and those that choose to have secure mail to the device can follow the standard.

c) Add some level of user authentication so that each time the device is turned on, or at a configured interval, the device prompts the user for a password.

BENEFIT: With several recent events security has become a key concern of RAM. It may be important to quickly meet these concerns head on with encryption.

STATUS: The addition of compression can be made to the existing product if the RAMFirst gateway is used for the back end. It is unclear if Fountainhead in Ardis has any encryption options available.

FACTORS: The inclusion of encryption to the current product would be useful, the enigma shareware product is so simplistic it could be considered non-encryption. However it is already in the gateway and is therefore easy to implement. The addition of any other end-to-end encryption methods would be difficult because RIM would need to find partners to do the server component.

Message Compression

FEATURE: Currently there is no compression in the Inter@ctive Pager product in either direction. Receiving compressed data would be much easier then sending compressed data with the current 'fixed' message sizes. This is because it is impossible to know how much data to allow the user to enter to fit into one network packet when compression is being used.

a) Add the compression protocol currently being used in the RAMFirst gateway. Add this compression to the HP-ID 4 protocol. **Note** : Performing compression on 'sent' packets using HP-ID 4 is very tricky and a limit of 512 bytes of input data would still be maintained.

b) Add support for a higher-protocol like Ramparts Transport Protocol (RTP) or Interactive Paging Protocol (IPP) and use compression on top of this protocol. This would solve the sending of compressed data problem by allowing a large message that did not compress into one packet to be sent in multiple packets as 'one' message.

BENEFIT: Compression lowers network traffic from the network operators point of view, reduces cost by transmitting less data, and increases overall throughput of data and effect transmission speed.

STATUS: Compression has been long absent from the Inter@ctive Pager. The problems of not having a transport protocol has created an awkward situation. It is possible to implement the current RAMFirst compression method (LZW) and continue to restrict the user to inputting only 512 bytes for Mobitex messages. In DataTAC it is unclear what compression methods, if any, are available.

FACTORS: Using LZW within the Pager, just as it is used within RAMFirst gateway, could be difficult. Initial investigations proved inconclusive as to how much memory and space the compression method needed. The people who developed the code for the gateway simply pick up some freeware from Internet and used it as is, without understanding it. RIM has sponsored a Masters student to develop a compression algorithm that would be Pager friendly, in terms of space and speed. This alternative method is better than LZW and must be considered.

Message Origin and Transport Options

By far of the most important areas of definition is how to get information into any RIM Pager. Getting third parties to start development these areas will produce the most important net gains. RIM has recently discovered that having a transport within the Pager products would be very important. However, we do not want to select one single companies transport, and are finding that all the existing transports are too large with too much overhead. This has prompted the new NBS definition for Mobitex and DataTAC, which we are considering placing into all of our product lines.

One area that has been purposely left out of this section is the option of implementing the RAM Parts Transport Protocol (RTP) and the Interactive Paging Protocol (IPP). RAM Mobile Data, thus far, have rejected NBS, have frozen all product lines based on RTP and have already gone ahead and ported IPP to the Pager. Therefore the document will not investigate these alternatives since they may not be viable development alternatives.

Messages from POP3 Connection

FEATURE: By working with a company like Infowave messages could be retrieved from any corporate mailbox and delivered to the Leapfrog. This can be done using a POP 3 access, as done by InfoWave's Golden Retriever product. POP3 does not require that the user's mailbox be moved, or that two mailboxes exist. The POP3 server effectively reaches into the user's account and gets and sends mail messages.

BENEFIT: It is probable that one way to successfully sell pagers would be to deliver a 'cost effective' corporate solution. The Infowave Gold Retriever product is a network or corporate solution that provides a very cost effective method for delivering mail. Microsoft is beta testing and using the product today with their Win CE products despite its security risks.

STATUS: Currently the Inter@ctive Pager has no transport. We are considering placing an NBS-like layer into the device, but no single company has appeared who is willing to implement our recommended fragmentation method. Bellcore has ported their transport and parsing engine into the product, but this is very specialized to Bellcore and runs very poorly over Mobitex. The NBS transport will be placed into Leapfrog and it should not cost much space in the product.

FACTORS: One of the biggest limitations is that POP3 is not secure and it is not likely that a secure POP3 will be implemented by any third party in the near future. Most companies doing this type of mail are moving to a direct MAPI solution. The current Golden Retriever product uses the RAM Parts gateway to perform the POP 3 mail pickup and delivery. Therefore if RIM wanted our Pager to work with the existing service we would have to implement RTP into the Pager, unless Infowave was convinced to do all the work themselves, both transport and mail application.

Messages from a MAPI Store

FEATURE: Messages could be extracted directly from the corporate 'Exchange Server' database using a product like RAD-Exchange. Currently RAD-Exchange has been purchased and is being released by Wynd. Other MAPI products soon to hit the market include Intra's Fountainhead, Ericsson's EVO and soon Infowave will be releasing a direct MAPI product. Any of these servers could act as a message redirector for a RIM Pager product. Some of the additional features can be implemented in this area include:

a) Coming up with an end-to-end solution with one of these companies would inevitably lead to implementing a transport-based solution, like NBS. Using a transport would allow larger messages to be delivered to the pager.

b) Many of these services will be implementing a 'message preview' option as part of their solution. Message preview allows only the message header to be delivered, and thus allows the user to act on each message based on the header information and message size. The user normally can delete the message, re-route the message or ignore the message.

c) Each of these services have a host of other features which cannot be all addressed here. Areas like multiple destinations, auto-forwarding, auto-reply and message translation as part of the base Microsoft Exchange Server product.

BENEFIT: The RAD-Exchange method would offer the best cost effective 'corporate-based' solution. The MAPI-based mail method gets each Pager user directly to their desktop mail, thus encouraging the 'attach-of-the-desktop' model. This method also offers the best method for increasing security options within a large corporation, where going through a network-based mail gateway is an unacceptable security risk and an additional cost for very little value. It is RIM's belief that having a message preview option is key to a successful mail product. When this feature was developed in RIM's mail products, it is felt that best way to achieve 'cost effective', wireless messaging was by combining a **header push / message preview** design model.

STATUS: Both header push and message preview could be achieved with collaboration with any of the MAPI based mail gateways or the Golden Retriever product. Currently there is nothing in the Inter@ctive Pager product that performs this function, most of it would have to be developed by someone.

FACTORS: The addition of a transport or of header preview would not be a large change. Naturally the hope is that by adding the NBS transport this will solve one major problem with working with these gateways. As for header push, headers could be treated like mail message and when requesting an action on a header the application simply needs to cross reference the original message. Naturally this entire concept can be abstracted for any message store, or even database product. I am leaving this issue as a 'massive' virtual market niche that will be and is being addressed.

Alternative Pager Applications

One of the goals for the Inter@ctive Pager is to encourage a myriad of applications to be developed for the Pager. The majority of these are likely to be vertical market solutions, but the hope is that many companies create a service offering and build many interesting horizontal applications. The following applications are some that RIM has targeted as important and should try to encourage their development.

Real-time Browsing Support

FEATURE: People with the Pager would like to browse web sites on the Internet and get at important real-time information when carrying the device. The previous statement has been made many times, but is it a true statement? The basic feature in question is the ability to enter an identification code or web site address and get information that is important to you. The following are the different ways this can be accomplished:

a) Add support for an HDML 1.0 message stream from an HDML-based gateway into the Inter@ctive Pager and Leapfrog. This product is currently being developed by Bellcore and RIM for the Inter@ctive Pager, which RIM will complete by the middle of the summer.

b) Add support for an HDML 2.0 message stream. This document has just been released by Unwired Planet (UP) and will be brought to the W3C Standards committee for the Internet. This protocol and parsing engine adds further options to the HDML 1.0 version, including graphics, and it would be an interesting product to target. Working with UP directly could help make this a more 'real' alternative.

c) Add support for direct HTML page browsing into the Pager. This would require working with a company that has an HTML proxy product, like ST Mobile Data in Singapore to create an end-to-end product. Obviously a lot of work would be required to create a 'unique' scrolling technique to look at the information, and all graphics would be filtered.

d) Add support for Nokia's HDML alternative product. This product is like HDML but Nokia didn't trust UP to bring their solution to the standards committees. The amount of market share for this product is unknown, and therefore the future content is also unknown.

e) Add support for Wink's ICAP standard, which is being heavily promoted in the TV set-top box market. ICAP is similar to HDML 2.0, with graphical support and an 'object' class orientation. The amount of market share of this product is unknown, similar to the problems faced by Nokia.

f) Create a browser based on the North American Presentation Level Protocol Syntax (NAPLPS) video-text standard. This standard offers many strengths for low bandwidth transmission for 'graphical' based screens. However, the standard is very old and many limitations exists. Therefore it is likely several strategic changes would need to be made to the native NAPLPS approach, resulting in creating something that could be called a 'new' standard.

BENEFIT: Adding support for Web Browsing has long been a goal of the product. With the planned development of a version 1.0 HDML browser, RIM would be taking only 'one' step in this direction. The need or demand for this is still unknown, **many** companies are trying to generate and test the market with such a concept. The biggest promoter of this wireless browsing concept is the CDPD network. In this case they have a large bandwidth which does affect the viability of such a service.

STATUS: The HDML product in a preliminary stage should be complete by the middle of the summer. However this will not be integrated with any E-Mail based application running on the Leapfrog or the Inter@ctive Pager. RIM is also planning to transition the software, with source code, to Bellcore once it is complete. RIM does not want to get into the situation of building and determining which version or type of browser to create for the marketplace. This would discourage other application developers from building such products.

FACTORS: Like so many new features, this is important if the market deems it to be important. Without solid market backup it is unclear how much effort should be put into pushing this application development. Marketing research is required to determine i) who is the dominate player, ii) who will have the best content in terms of web sites and iii) who is willing to work in the Mobitex and DataTAC area.

Schedule Tracker

FEATURE: There is no schedule based application or sub-system on the Inter@ctive Pager today. Therefore any work done in this area would be new. Once of the main reasons that a scheduler has not been considered for the Inter@ctive Pager is the lack of screen space. Typically to view any quantity of information more space is required.

The combination of an Contact Manager, E-Mail Assistant and a Scheduler would place the Pager product line closer to true PIM system. There are several ways to approach the problem, inside and outside of RIM. Further detailed and technical information on how to develop a Scheduler is found in **Appendix A**. Some features, and different types of Schedulers could include:

a) A stand-alone scheduler on the Pager that allowed a user to enter schedule events locally through the keyboard.

b) Continue to use the 'on-board' configuration and synchronization methods available in the Inter@ctive Pager today. This means I can schedule meetings and send 'schedule-type' messages to people from the Scheduler within the Leapfrog. This would naturally work best with other users of the Leapfrog that can 'understand' the message format.

c) Creating a standalone configuration application, just like the one that is available for US Robotics Pilot PIM product, that allows a user to enter information on a full screen and 'get it to the device' (See Appendix A for more information).

d) Create an automatic schedule updating product, unlike most other product on the market. This product would connect directly with 'all' major contact and scheduling product like, Schedule+, Outlook and Goldmine. If a change was detected, or a Synchronization button was pressed, then the product would automatically pull out the information and send it to the device wirelessly.

BENEFIT: The concept of attacking the desktop with a Desktop Partner would require this kind of application and this level of seamless integration. Naturally a customer would be WOWed to see automatic or semi-automatic update appear magically onto his Pager device. Most executives would use the device with their executive secretary updating their information on the fly. Having the ability to work out schedules with other Pager users '**may**' be a very interesting marketing feature.

STATUS: No scheduler has ever been available for a RIM Pager, although the Inter@ctive Pager does have the ability to set a one-time alarm, and turn itself on and off at given times.

FACTORS: We are planning some 'form' of configuration tool for Leapfrog. At this point input from Marketing is very low and we will be defining the product very soon with or without additional input, apart from the current information which is to 'make it like the pilot's tool'.

Import Facility

FEATURE: The current import facility has two modes today, the first is 'active mail', which is a method that requires manual intervention when moving a message into, or out of, the received mail area to the address. The second allows a standalone configuration utility to be used for updating the device either over the air or through the serial port of the Pager. This is considered a 'trusted' server component and performs a direct update of the address book or configuration options. This basic import facility can be updated in many ways to enhance and improve its capability, the options include:

a) Create a more sophisticated synchronization tool that allows information to be exported from Schedule+, Outlook, or Goldmine and imported into the synchronization tool; similar to the Pilot software.

b) Improved the Import facility in the Leapfrog so that messages that were sent would not have to have a 'rigid' format for the data. This would allow the user to apply a template onto the received message to identify 'important' fields to be imported.

c) Ability to be invoked as a 'quiet' sub-system that simply auto-detects the message format and imports the correct contents to the correct database.

d) Extended the ability to 'export' address book entries into the area of schedules. This would allow two pager users to agree to a scheduled meeting.

BENEFIT: The import facility would allow a user to quickly and easily create a complete address/contact book, phrase book and schedule information. It also allows for the easy creation of synchronization software to be built that can send the correctly formatted information to the pager. The additional features simply make the process of getting the information easy; i.e. the user doesn't have to re-input the data again. Improving the Leapfrog's import facility would allow a wider variety of formats to be received and thus increase the people who could send me address information. The import facility would allow seamless integration to the desktop, similar to what the Pilot product from US Robotics is doing today already.

STATUS: Some of these features are currently available in the Inter@ctive Pager product, and a simple piece of synchronization software also exists for the Inter@ctive Pager.

FACTORS: The redesign of the software for the Leapfrog will provide an ability to treat each area of the software as a separate entity. This is true for the address book today, which allows the address book component to call the messaging component to send or receive an address book entry. Similarly the scheduler would be able to call the messaging component to perform its synchronization technique.

Current Pager Enhancements

Within the basic pager environment there are minor changes and enhancements that could be made that could make using the Pager easier. Some of these changes could be quite difficult to implement and given that we are promoting the development of alternative applications on the device they may not be worth the effort. These changes are primarily focused on local 'user interface' issues, but also extend to some functional enhancement. Some of the changes in the Application area, do not require the addition of the RTP protocol used within the RAMFirst gateway.

Application Behaviour

FEATURE: For overall application behaviour there are several features in the RAMFirst gateway which are not being used today. Additionally several advanced features have been identified that could be changed within the Leapfrog to make it a better product. These features include:

a) When receiving only the first 512 bytes of a message from RAMFirst, there is a simple mechanism that could be implemented to get the next portion of the message, i.e. a '**MORE**' button. This method is independent of whether the Pager has implemented RTP.

b) It is possible with RAMFirst to put multiple destinations into an outgoing message, causing it to be routed to many recipients. This does not require RTP to have been implemented.

c) It is possible with RAMFirst to forward a message to another destination, once the first 512 bytes have been viewed. The message could be forwarded to a Fax machine, or to another e-mail address. This does not require RTP to have been implemented.

d) It is possible with RAMFirst to set up an auto-forwarding rule, so that if the destination user is travelling or out of coverage for a long time, mail is auto-forwarded to another location. This does require RTP to be implemented.

e) It is possible with RAMFirst to have all incoming message to also be copied and forwarded to another mailbox location. This helps solve the two mailbox problem, by ensuring a message goes to your LAN mailbox and your mobile mailbox. This requires the RTP to be implemented for this feature to be turned on.

f) The concept of getting 'time' for the device from the network time stamp. In Mobitex this would be trivial to implement. The Pager would have two times, the Home Time, and when the time zone changes due to travelling the Pager would also have a Local Time.

g) Further improvements in setting up Application Options would always be a good area to work on. Some areas include:

- defining what the 'default' mail view will be for the device; current when you change the view that view becomes the default
- adding options for auto-fetching 1, 2 or X more packets of a message held at the gateway; this saves having to press a 'More' button several times
- setting up import filters or other automatic methods for getting new information into the device

BENEFIT: Each of these features can be evaluated based on the marketing feedback for the device. Now that the Inter@ctive Pager is getting into 'real' customer's hands it is possible to get feedback. The result of such market research would help determine if there are hot development areas to be exploited. Application options gives the user control and customization over the product. Expanding this area is usually seen as a good thing, as long as the default behaviour is sufficient for most unsophisticated users.

STATUS: Most of these features are not in the Inter@ctive Pager today, therefore each one must be evalutated on its own. Since there is the ability to change Application Options is should be easy to add more options.

FACTORS: To differentiate the Leapfrog from the Inter@ctive Pager, it appears obvious that we should add in the few simple functions we can to improve the product, without adding RTP or IPP protocols.

Message Exchange - Viewing

FEATURE: In the area of message viewing there are several feature choices that are possible, including:

a) Expand the database engine and allow the user to define their own view. For example I want a view that includes only "all message received last week from Mike Lazaridis", or I want a 'NEW Message ONLY" folder. Effectively this would be similar to creating folders, only the user does not have to place the mail in the folder and mail can be part of several views simultaneously. The default three views would be 'all messages', 'Outbox messages' and 'Inbox Messages'.

b) Mark a group of messages for deletion. Currently a pointer to the current message is maintained and all operations are performed on this message. Or drag the 'roller device' over a group of messages to mark them for deletion.

c) When using a **header push** model the message viewing screen would contain only message headers. These headers would include the source of the message, the subject of the message, the size of the message and the time of the messages. Additionally the priority of the message and part of the body could be shipped across to help the user make the decision to spend the money to receive the message wirelessly or not.

d) Place an '**integrated**' HDML browser directly on the device. This way when HDML messages are received and opened, the HDML viewer would be invoked to display the HDML decks and cards. This would mean that two message viewers would be present at once, and based on the message format different viewers would be invoked. Even more important is that if HDML messages were received and the HDML viewer is the currently active task then the messages should be delivered directly to the program and make immediate screen changes; this is required to facilitate 'near real-time' interaction with a web site.

BENEFIT: The current viewing method is one of the most common and always gives the user a clear idea of what has been sent and received and in what order were the messages exchanged. The problem is that very often the one view gets very full and it gets hard to find messages. With the addition of creating separate views for Inbox and Outbox, the user can more easily find a given message. Naturally, as anyone uses a mail system, it becomes important to classify messages further and be able to quickly find a message from a given day, or from a given person. The current Inter@ctive Pager also has the concept of a 'saved' message area. Messages can be moved into this area for long terms storage and retrieval. This saved area can be used as a 'special folder' for important messages.

A more complicated change would be to allow the user to define their own message view. Often when using the Pager a user gets a large number of legacy messages stored. Delete these message one at a time can be very time consuming. It would be nice to be able to mark a group of messages and delete them all. Currently the Inter@ctive Pager can delete a group of messages after 'N' days.

As already mentioned the **header push / message preview** design could be a fundamental selling point of the device. It could be a point of frustration if there is a delay each time I want to open a message since it must be fetched from the server each time. An integrated HDML viewer means that I can seamlessly access 'certain' web sites. Which web sites I can access and how quickly I can receive my information could make this feature great or useless.

STATUS: The basic 'all messages' view, 'Inbox-Only' view and 'Outbox-Only' view are complete and working on the Inter@ctive Pager today. The other more complicated viewing option, described in (a), is not completed and had been planned for the Leapfrog but are currently on hold. Feature (b) above is also not implemented in the current product.

FACTORS: Bellcore has already been talking about an 'integrated' HDML view and mail viewer. In Bellcore's case both viewers were simply HDML decks. RIM is not generally in the business of doing applications that are this sophisticated, and this level of integration may require it.

Address Book - General

FEATURE: The additional features that could be added to the address book area include:

a) Allowing the user to add their own fields and give the field a dynamic name of their own choosing.

b) Allow the addresses to be classified into different areas or views. This could be used in conjunction with the changes in the Messaging area described in change (a) above. This would allow the user to have a Business, Personal or Total view of address information.

c) If messages were detected as being address-based synchronization messages from Schedule+, Goldmine or Outlook, then it would be automatically pulled into the address book's database.

BENEFIT: During message creation the address book saves time in constructing the destination address. When authoring a message, having an address book is important since most destination addresses are not memorized and are too complex to remember. Additional contact information can be stored on the pager, thus making it a simple contact manager as well. Adding the ability to create dynamic fields simply makes the address book more like a contact manager. Adding the ability to classify the addresses makes the address book easier to maneuver and it takes less time to find an address the user is looking for. The concept of attacking the desktop with a Desktop Partner would require this level of seamless integration.

STATUS: This feature is currently available in the Inter@ctive Pager product. However, it is currently not possible to include more then one address as a destination on a message. Some amount of auto-updating is already supported by the Inter@ctive Pager today

FACTORS: This has already been discussed somewhat in the Import and Synchronization sections. The address book is already very flexible I personally doubt that much user benefit can be added.

Phrase Book

FEATURE: The additional features that could be added to the phrase book area include:

a) Allow the phrases to be classified in to different areas or views. This would take advantage of the changes described in the Messaging section under change (a) above. Effectively phrases could be grouped into categories for responding to personal messages or messages from the office.

b) If messages were detected as being phrase book updates, probably directly from the synchronization tool, then it would be automatically pulled into the phrase book's database.

BENEFIT: To assist in the creation of mail responses, and self authored messages, the phrase book can be used when sending standard messages. This area is common across all Mobitex server options, it is not likely to change based on the back-end server that is selected. The concept of attacking the desktop with a Desktop Partner would require this level of seamless integration.

STATUS: This feature is currently available in the Inter@ctive Pager product. Some amount of auto-updating is already supported by the Inter@ctive Pager today

FACTORS: Changes and improvements here would be more valuable for the Kermit environment. I don't think that there is much point improving this area as it mostly used by virtual solutions, and since virtual solutions are generally done from scratch they would not be using the Phrase Book included in the base application.

Expanding The Pager's API

One of the more difficult decisions being considered is 'extending' the current Pager's API to include a richer set of features. It has been argued that the developer's like the current simplistic API, and it should not be changed. Others have argued that developer's are asking for more assistance in the creation of elegant forms and User Interface applications. If we consider that the current application took RIM 6 months to develop, and get working just right, and that several 'non-documented' API calls are being used to perform some of the magic within the current application, then it is worth considering that adding to the API is worth while.

NBS Changes

FEATURE: The additional API calls for NBS will be easily added without affecting the existing API functions. With NBS comes the concept of a TCP/IP port. Ports are simply semaphores used to co-ordinate two tasks when communicating end-to-end. Naturally a program wishing to use the NBS transport layer will use functions like:

a) Listen : Allows an application to 'register' or become a server on a given port. Multiple applications are able to listen simultaneously and perform complex functions across a 'common' port end-to-end.

b) SentTo : The send to function will allow the sender to specify a destination, some message parameters and a message length up to 128K in Mobitex and 512K in DataTAC. Naturally from a hand-held device these sizes are more than enough to perform any function. The NBS layer uses the correct reliable fragmentation method described in the NBS for Mobitex and DataTAC document from Research In Motion, Version 1.6.

c) ReceiveFrom : The receive from call allows an application to pick up messages or datagrams as they are received. A datagram is composed of one or more fragments, which are reassembled and given 'reliably' to the application listening on a given port.

BENEFIT: For companies wishing to create end-to-end applications the requirement to send more than one packet is often their first requirement. The presence of an NBS protocol layer, that can be optionally invoked, is viewed as an important feature. The main benefit is that it could save developers months of time porting or writing their own transport layers. It allows RIM to 'partner' with wireless middle-ware developers if they put NBS into their protocol stacks.

STATUS: This feature will soon be coded into the Inter@ctive Pager and will be ported to the Leapfrog.

FACTORS: There has been a lack of feedback from Netteck, Ardis, Racotek, and Integra regarding the NBS for Mobitex and DataTAC specification. It may be possible to force them to use the protocol, based on the Pager having great sales, but this is a gamble we are going to play.

New UI Engine

FEATURE: The most common request for the Inter@ctive Pager from developers seems to be for a rich set of User Interface (UI) API calls. Currently there is no concept of list boxes, scrollable regions, menus or edit boxes. Each developer is forced to create these for every product developed. Based on the low-level LCD routines, this work has already been started for the Leapfrog and the current application has been written to use this new UI Engine. The new UI Engine presents an 'Screen Object Class' to the application with the following features within the object:

a) Edit boxes - this is a multi-line element used to input data (e.g. used when user is composing a mail message). This performs word wrapping.

b) List boxes - this is a multi-line element used for presenting a list of items to the end user (e.g. the list of mail messages in your inbox)

c) Label boxes - this is a single line element used, for instance, as title lines for other elements

d) Choice boxes - this is a single line element is when presenting feature options to the end (e.g. key tone options - long, short, off) Cursor keys are used to choose between the options.

e) Line edit boxes - this is a single line element. If the inputted text is longer than the display area, the text will "slide" from left to right depending where your cursor is. Inputting long internet addresses is an example of its use.

f) Menus - this is a drop down menu used

g) Dialog boxes - user responds yes/no

h) Status Boxes - same as dialog but goes away on its own e.g. Receiving Message

i) Bitmaps - which allow for bitmaps to be placed on the screen, similar to what is used today in the Inter@ctive Pager

BENEFIT: RIM would like companies to be able to develop 'sexy' and useable applications easily and quickly. By offering an extended API for the UI we can accelerate the development of applications and create a larger ground swell of developers promoting the product. Word of mouth in the development community is very important, especially since the wireless developers are so few.

STATUS: This UI Engine will soon be coded and available for the Leapfrog product. RIM should probably either make it part of the API or release the source code for other companies to use.

FACTORS: One of the discoveries of developers is that the current Pager application is much harder then it first looks.

Complete Application API

FEATURE: A long standing goal of the Pager development team has been to create an extensive API including the 'existing features' of the application. The major features of Message Display, Address Book/Contact Manager, Scheduler and Transport (NBS) would be available for the program writer to use. Thus an application would simply need to perform the following steps:

a) Receive 'messages' from the NBS transport or the network.
b) Decode the particular message and verify it's meaning.
c) Call the Message Display, following the API format and guidelines to display important information to the user.
d) Call the Address Book manager to save important 'addressing' information for sending and referencing or perhaps for contact type purposes.
e) Call the Scheduler for setting special alarms or priority information, including actual scheduled meetings, next 'call' information, time limits on current call, time limits to get to the next call, etc.

BENEFIT: To promote the concept of quick and 'very' easy development of application on the Pager this would give the development community a major step in that direction. RIM is hearing about many companies that are simply sending their information as mail messages to the base application because they like it and it works. Therefore it could be assumed that many companies would use these components in a 'generic API' form if we provided them as API extensions.

STATUS: Some work has been done in this area and continues as the Leapfrog application is written. The Address Book and Transport will have this model and it could be possible to invest more time and make all the major components a standalone API that is callable.

FACTORS: One of the discovers of developers is that the current Pager application is much harder then it first looks. This offering is even more powerful than the last in its level of completeness.

Adding Racotek's Key-Script Language

FEATURE: To address the functionality that Power Tools is going to promote, there is the option of adding Key Script to the product. Key script would provide the following features:

a) R

BENEFIT: To add value to the customer, and allow for many more integration methods, RIM has the option of adding a strong product into the marketplace.

STATUS: This would be a completely new effort and require several months of effort.

FACTORS: Racotek suggested this and have agreed that if we wanted to use it they would make it a public standard. Racotek realizes the need for public disclosure and are being upset by RAM with Power Tools. It would also appear that many concepts being used in Power Tools was taken from Key Script directly.

Appendix A : Wireless Host Access

One of the biggest stumbling blocks for new companies wanting to add end-to-end services or products with a RIM Pager is the challenge of adapting their Host software to talk over a Wireless Data Network. Some of the new services announced by RAM and Ardis for Internet connectivity will be a stepping stone to solving this problem. The appendix tries to describe other alternatives for the future and how companies, like Starfish, could create synchronization software of the RIM Pager family.

Internet Access Phase 2

Fixed link access to the Mobitex and DataTAC wireless data networks has recently been expanded to allow Host systems to connect via the Internet. This important step has been central to reducing the cost of Host access, simplifying the access method and making pilots and implementations easier to justify. The current Internet connection method is a "Stage 1" approach, that addresses medium to large size companies with fixed Internet access and a fairly strong corporate view of wireless.

Now that Stage 1 is under way, it is time to look at a "Stage 2" implementation that goes beyond the dedicated TCP/IP connection to a dynamic TCP/IP access based on an individual mobile device. What this means is that validation of the Host access is **not** done based on the source IP address, but based on another validation method as recommended in this document.

Internet Access Details

The Phase 2 wireless Internet model is based on a dynamic coupling of the Host and the Internet bridge to the wireless network. This means that each time a new Host wants to connect through the Internet Gateway to delivery packets to a mobile, it would **not** have to register and be manually configured by the network operator.

The approach is that for *every* Inter@ctive, Leapfrog and Kermit Pager sold an automatic Internet Host access configuration would be added to the Internet Gateway. Sold with the Pager, or purchased separately from a third part, would be a 'special' configuration disk for talking to the Pager. This disk would contain Host-based software that would attempt to reach the Pager via the Internet. If there were 100,000 Pagers sold and 100,000 Hosts that may want to connect over the Internet, only 100 of them would probably be active at one time. RAM or Ardis do not have 100,000 fixed access ports to supply for these 100,000 units, thus a dynamic coupling method is required. This means that all 100,000 Hosts should be know and configured, but that only 100 'could be' connected at once.

To facilitate this coupling, the Host (configuration/synchronization software) would be initiating the flow of data to the mobile by opening the connection to the wireless network as it does in phase 1. This is necessary due to the fact that the mobile can never know for certain what network address the Host number has been assigned, or whether the Host is currently up. After the connection is opened, the Host would be validated using a higher-level authentication method. This method would be either through a specially constructed network packet, like a DataTAC 5000 Authentication Packet or a Mobitex Personal Sub-Login MPAK, or using the new Point-to-Point Tunneling Protocol (PPTP) being standardized by Microsoft and others.

Internet Access Changes Required

This dynamic coupling may sound **too** easy, but with some minor work on the Internet access into Mobitex and DataTAC it can happen. The following changes are required to make the above scenario more possible:

- Each device that is packaged to be sold has an account and password registered within it. When a customer gets the device their profile has already been registered at the Wireless Internet Gateway for the network that has been selected.
- When the synchronization tool is activated for the first time the Wireless Internet Gateway becomes aware that a new customer is accessing a wireless device through the Internet. This does **not** cause billing to activated because the 'real' costs are tied to the device and registration of the device's purchase has started the mobile customer billing and activation process.
- The Host-based synchronization software that access the Internet for connection will use port 80, not some higher level port. Port 80 is normally used by HTTP, and is thus normally opened to all users on the Intranet

within a company. By using port 80 the customer will not have to go to the MIS department and make a special request to allow a 'new' connection through the firewall; even if these connections are only initiated from within the company. However this is not a guarantee, some companies like Intel have both an Internal and External firewall for the company. There are also software packages that will 'poke' at the firewall until they find a hole. This is becoming common among new Netscape add-on products.

- The user configures the mobile's network address with the Host synchronization software so that network packets can be addressed to the device. The first step for the conversation will either be to send a 'Configuration' packet, or simply to send the synchronization information. The configuration packet would be sent to provide the mobile with the currently assigned fixed address. This would be sent if a 'full' synchronization was requested, otherwise a one directional message transfer would take place.
- Finally, the Host synchronization software is password protected with an encrypted user password. This password is set to a default value found within the packaging of the device, and must be change by the user the first time it is used. This password is also maintained by the pager so that the original packaging cannot be copied and reused maliciously. However, the owner who knows the password can set up an additional site, like a secretary for assistance in synchronizing schedule information.

The goal of the network operator would be to create a Internet Access Database that would contain an authorization code for every Pager that was sold. The Internet Gateway code would allow a maximum of 'N' connections to be received on Port 80, before starting to reject connections. The Host Configuration/Synchronization software would ideally only run for seconds or minutes and finish its transfer very quickly. Any resolution of synchronization conflicts would be handed after the connection closed locally. Conflicts on the device would take the 'older' of the changes and the Host would be able to over-ride this and then download the change again.

The conclusion is that companies may still have to consider the slow-bandwidth and long latency of the wireless network when building or adapting their software for Mobitex or DataTAC. The question that could be asked is, if companies have to change their software to make it wireless friendly, then why not simply go directly to the wireless network? The answer is ubiquity, cost and ease of use. The customer wants to use this synchronization tool anywhere and anytime, without hassle and failure.

Alternative Connections

When RIM does its configuration tool it will probably support four different methods for delivery the information to the device, these options will include:

- send the information directly through the COM port of the computer into the modem directly
- send the information through a RAP conversation to a radio modem wireless to the Pager
- send the information through an Internet Tunnel to the Pager
- send the information through MAPI as a mail message to the Pager

The last case may be the hardest since it may result in the configuration tool having to create only a 512 byte message to hold the information. Also the Pager goes through a 'manual' import process when receiving address book or schedule information from a mail message. This security precaution is to ensure that unwanted people cannot affect my Pager without my knowing it.

For any company perform this operation these steps should also be considered.

Table Of Contents

Introduction 1
Version Summary 1
Technical Summary 2
General Requirements Overview 3
Stage 1 Requirements 4
User Interface (UI) Sub-System 4
Options Sub-System 5
E-Mail Sub-System 7
Address Book Sub-System 10
Phrase Book Sub-System 11
Additional Sub-Systems 12
Desktop Support Software 14
Basic Desktop Support 14
Advanced Desktop Support 14
Stage 2 Requirements 15
Stage 3 Requirements (or later) 17

 © Research In Motion Limited, 1997

RIM CONFIDENTIAL DOCUMENT
Classification: COMPANY SECRET

Document Title:	LeapFrog Requirements Specification
Document Revision:	Version 1.0
Document Date:	November 25, 1997
Document Reference Number:	ENG-LEAPSPEC.DOC

This document is issued for signoff to:

Recipient Name: Mike Lazaridis
Recipient Company: Research In Motion
Sign Off Signature:

Recipient Name: David Werezak
Recipient Company: Research In Motion
Sign Off Signature:

Recipient Name: Michael Barnstijn
Recipient Company: Research In Motion
Sign Off Signature:

Recipient Name: Gary Mousseau
Recipient Company: Research In Motion
Sign Off Signature:

Signoff of this document indicates agreement of the specified version 1.0 functionality. Once signoff occurs there is a formal method for adjusting the functional defined in this document. By using an Engineering Change Request (ECR) it is possible to adjust and change the specified functionality in this document.

© RESEARCH IN MOTION LTD. 1997

Proprietary and Confidential

Introduction

The decision to create the PageMail Server has left the Leapfrog's application software deadlines extremely tight. In an effort to reduce risk of failure, and risks of producing the wrong software, the Leapfrog group has produced this document in an attempt to describe the desired functionality of the Leapfrog application, something not commonly done at RIM. It is hoped that the sign off process will encourage all the key individuals to review and carefully think about what the Leapfrog is, and what Leapfrog should be able to do.

The Leapfrog Requirements Specification is designed to be a detailed summary of what Leapfrog will be able to do in Version 1.0, and a terse summary of what Leapfrog can do in Version 2.0 and 3.0. This document's goal is to describe the functionality of the software, not present specifics on the user screens and visual presentation. There are several other documents that can be read if the reader wants to review exact screen presentations. Additionally, the scope of this document is to describe 'what' the user can do once they purchased a pager and are about to use it.

Version Summary

Given the short timelines for producing a functional device, the list of requirements has been broken up into a staged delivery. The first stage device will be completed and tested before the second stage is started. Expected delivery dates for stages are:

- Stage 1 – Basic Device Functionality – At this stage, the leapfrog is an Inter@ctive Pager with everything the Inter@ctive pager does, but better, and in a smaller package. Some improves to PageMail to support

  Alpha Available – Jan 15th
  Beta Available – Late January/Early February
  Completed – Late February

- Stage 2 – Advanced Functionality – Adding forms, a calendar, and other extended features to be determined from marketing feedback and internal requirements. Some back-end work for forms support and SWAP interface within the PageMail server.

  Alpha Available – Feb. 30th
  Beta Available – Mar. 15th
  Completed – Mar. 30th

- Stage 3 – Further Device Improvements – Public Key Encryption, advanced and as of yet undefined extra fancy features. Probably based on marketing and internal requirements.

  Alpha Available – ?
  Beta Available – ?
  Completed – ?

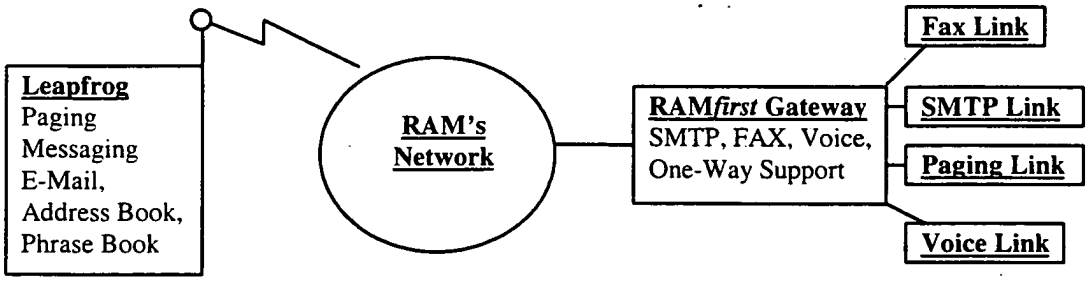
 © Research In Motion Limited, 1997

Technical Summary

The basic Leapfrog application software must be developed to communicate with RAM's RAM*first* Gateway and RIM's PageMail Gateway. This will be done by developing **two** different applications and reusing as much code as possible in both applications.

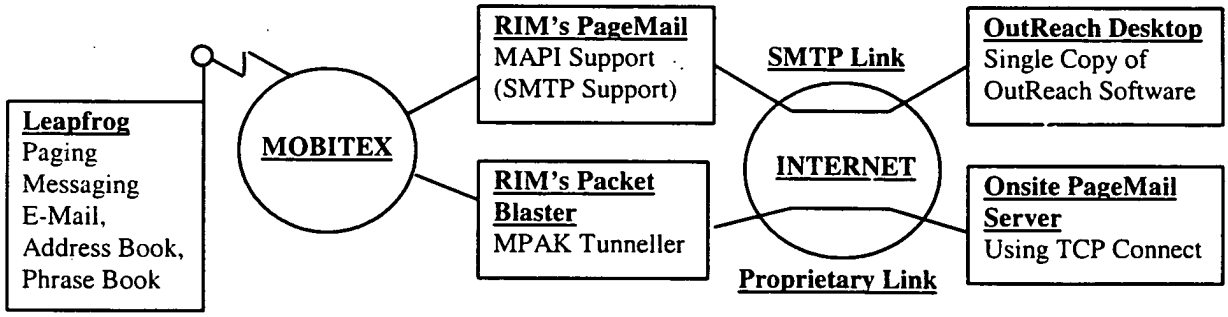
All Leapfrog devices provided to RAM will communicate via the RAMParts Transport Protocol (RTP) to the RAM*first* Gateway. The RTP transport protocol is very similar to the MDP transport and therefore they will replace either other nicely. It will provide delivery of pages and messages beyond 512 bytes, and RIM will be adding a configuration command that will allow the user to set the maximum packet size that can be delivered to the Pager.

The following simple diagram depicts the end-to-end solution:



Connecting to the RAM*first* service will provide the customer with a range of services, depending on which services are connected to any given RAM*first* Gateway. This document will not discuss the services specifically, but will describe how the user accesses those services to achieve the benefits they offer.

The transport method used within the PageMail server is RIM's MDP method. The PageMail server can either be run directly connected to Mobitex via X.25, or via the Packet Blaster using TCP/IP over the Internet. Additionally the PageMail server can also support communication directly between the OutReach product and the Leapfrog.



PageMail will not initially support FAX or other Exchange Server transport methods directly. However in time there will be a series of additional routing option supported.

 © Research In Motion Limited, 1997

General Requirements Overview

Due to delivery schedules, release 1 of Leapfrog is primarily focused on RAM's requirements. As such release 1 only contains a portion of RIM's overall requirements for the Leapfrog. The following is a quickly summary of message delivery and back end support:

- RAM's application will have a minimal RTP protocol implementation with no header support, but support for one configuration packet
- RIM's application will use a MDP protocol with a compressed MIME (CMIME) data format and support for one configuration packet
- The main purpose of the configuration packet sent by RAM's application is to change the maximum packet that can be exchanged between the Pager and the final destination
- The purposes of the configuration packet sent by RIM's application is to change the maximum packet size and provide a signal to OutReach to start mail redirection
- RIM's application will also have proprietary compression and encryption algorithms and will use a private key from end-to-end
- The user will have the ability to configure a range of notification behaviors, depending whether the device is in or out of the holster.

The following are also considerations for back end support:

- Version 1 of RIM's application will work with PageMail as it stands today with minor changes for CMIME support
- Version 1 of RIM's application will work with OutReach with improvements and enhancements to OutReach, these include:
  - Richer integration with Outlook and MAPI
  - Ability to redirect mail, calendar and contacts
  - Support configuration packet for setting maximum packet size and redirection turn on

The following are major functional components that have been discussed but will not be present in Version 1.0 of the Leapfrog or PageMail/OutReach solution:

- Complex table support in the User Interface (UI) Engine
- Multiple game support and a complex calendar application will not be present; these also require complex tables, which are not available until Version 2
- Having added CMIME formatting in Version 1 will allow the addition of forms and scripting support
-

 © Research In Motion Limited, 1997

Stage 1 Requirements

The following is a section by section summary of the functionality being planned for Release 1 of Leapfrog. The first two sections are common to all other sections, these are the User Interface and Options or System sub-system.

User Interface (UI) Sub-System

The following table is a summary of all UI calls within the UI class library. All calls are C++ member functions within the UI sub-system managing all screen and keyboard control. However, it is important to note that even though the UI Sub-system acts on most keystrokes, it does give the currently executing application first chance action on the key.

User Interface Feature	Brief Description
Lists	A List Box is a read-only list of choices that one item can be selected from and then dismissed
Menus	A Menu List is a read-only list of choices that many items can be selected from, and it must be manually dismissed
8 Line Mode	In the first Release 8 lines of text will always be displayed on Leapfrog's screen; as opposed to 4 or 6 lines
Dialog Box	A dialogue box is displayed as a notice or warning and must be dismissed by the user
Text Box	A read-only field with text data
Edit Box	A read-write field that allows full editing capabilities for information input
Choice Box	A dialog box that allows the user to make a selection between several choices
Multi-Choice Box	A dialog box that allows one line to contain multiple choice boxes
Status Box	A stationary informational box, like a title that does not move
Simple Tables	A simple tabular data set that allows the user to maneuver through a set of fields
AutoCaps Vs KeyRepeat	User can select between auto capitalization and key repeat; auto cap will take place by holding the down
Preferences Screen	The UI sub-system will expose a series of options through the Options and System menus.

 © Research In Motion Limited, 1997

Options Sub-System

The following table is a summary of all Options and System functionality planned for Stage 1. It is important to keep in mind that certain areas like Other Application Configuration and Backup/Restore involve implementation routines from other sub-systems. This is because the options is simply a single interface method for these two important functions.

Options	Description
Options API	The options sub-system will expose an API for other application to post special callable routines
Other Application Configuration	Each application that has options to be changed can choose to provide a route to the options sub-system for centralization
System Wide Backup/Restore	Each application that has a database that can be backed-up or restored can provide a routine to the options sub-system
Date & Time	There will be options for setting the time and data on the pager
Screen Saver/Clock	There will be options to determine the behaviour of the screen saver screen and clock mode screen saver screen
Device	There will be general device settings that can be viewed
Pager Status	The Pager status screen will display the state of the radio link and other system values (perhaps even memory availability)
Auto On/Off	Like the Inter@ctive Pager there will be Automatic On/Off options, including weekend operation
Switch To	The system menu will manage the 'Switch To' menu options for jumping between sub-systems
Holster/Non-Holster Notification Support	There will be configuration settings for guiding how the pager notifies the user when inside and outside the holster
Symbol Table	The options sub-system will manage the symbol table and the pressing of the symbol key across all sub-systems
AutoCaps ON/OFF	The Auto Caps behaviour will be implemented by the options sub-system ??????????????????????????????????????
MESSAGE SIZE & Redirection Turnon	This option allows the user send a configuration message to set their maximum receive size and to start mail redirection
TURN OFF	The options sub-system will manage the Turn Off keystroke as entered by the user
BACK LIGHT	The options sub-system will catch the back light keystroke and turn the back light on and off

From the user's point of view, the Leapfrog will act similar to the Inter@ctive Pager, except that they can select the notification mode differently when inside and outside the holster. These settings include:

- **Vibrator Only Mode**: This mode causes the device to vibrate when a new message is received.
- **Beep Only Mode**: This mode causes the device to beep when a new message is received.
- **Beep and Vibrate Mode**: This mode causes the device to beep and vibrate when a new message is received.
- **No Notification**: The user can run the device in silent mode so that it neither beeps nor vibrates.

In addition to being able to set holster and non-holster mode differently, their will be two levels of notification for when a new message arrives in and when messages remain on the device unread. This alert mode is a 'wake-up' to the user in case the first notification was missed.

????? WANT A LIST OF EVERY CONFIGURATION PARAMETER !!!!!!!!!!!!!!!!!!!!!!!!!

 © Research In Motion Limited, 1997

located in the Message View menu, not in the Options Sub-system. Global settings like alarm, contrast and others will be placed in a "general" Options Sub-system section.

The other main goal of the Options Sub-System is to combine the original system and application options into one options area. The Inter@ctive Pager has two separate sets of options, accessed through the application and the operating system; in the Leapfrog, these options will be combined. The sheer number of options will be reduced by moving options to their respective sub-systems and by carefully managing which options the user sees. For example, the Leapfrog may have an 'Advanced Options' switch that will be turned off by default.

The options the user will select from will include:

- **Time**: Allows the setting of system time, the setting of timers to turn the Pager off and on daily and the setting of a one-time alarm.

- **Device**: Allows the setting of contrast, key clicks, notify modes, security level and turning the radio on and off.

- **Information**: Shows the user general information about the device, like the current radio coverage, device number (MAN), Password security level, serial number, Mobitex Network and Base ID, O/S Revision and Software Revision numbers.

Message Options: Allows the user to set confirmation on message delete, delete period on old messages and Gateway addressing options.

 © Research In Motion Limited, 1997

E-Mail Sub-System

For every message that arrives in the user will receive the configured notification. The user will be able to immediately press a key and see the message causing the notification alarm. To distinguish between read and unread messages a special indicator will be shown on the list screen. Messages can range in size from 1 character to an undefined upper limit, somewhere in the 10K to 15K range for incoming messages. This can be configured by the user in either the options or Email sub-systems. The Outbox will be able to show a simple indicator as to what has happened to transmitted messages, i.e. pending, sending and sent. The following is a brief list of functions within the E-mail sub-system:

Email	Description
Message Interaction	Messages will be received with either RTP or MDP, depending on the Gateway used. Message list will have a title by date with folder being viewed
Messaging API	The message sub-system will export an API so that other sub-systems can send messages, i.e. address book entries
Attachments	The message sub-system will be able to handle a small set of attachments, i.e. address and phrase book entries. When no clear data is in the message the attachment is directly given to the correct application automatically
Filters	A simple filter can be set up within a folder - like show me all messages from a given e-mail address
Folders	There will be three static folders for mail, the Inbox, Outbox and Saved folders
Compression	When working with RIM's Host products all messages will be send and received in compressed format
Encryption	When working with RIM's Host products all messages will be send and received using encryption, with a key of 0
Reply To: field support	When working with RIM's Host products a 'reply to' field will be accepted, and used for addressing replies
Insert Key functionality	When reading messages the user can press a key (ALT I) to insert the e-mail address into the address book
Holster Functionality	When in idle mode, and the user pulls the pager out of the holster, they will be taken directly into the message for reading
Advanced Compose	When composing a message multiple TO:, CC: and BCC: fields can be entered for addressing purposes
Customize	The user will be able to customize how information in a folder is displayed; subject and time can both be removed or added to the list view (**NOTE Below**)
Message Management	Hundreds of messages will be stored but to avoid running into capacity limits message will be automatically deleted based on a configured time period; the saved message folder will not be deleted however as are the Inbox and Outbox

The Leapfrog will be capable of holding hundreds of typical length messages. The user needs to avoid exceeding memory 95% full, since this condition will cause excessive "garbage collection" and Flash erase cycles. To avoid this situation the Leapfrog will have several memory management features:

- Encouraging the user to set a maximum message size of between 4K and 10K bytes.
- Automatically deleting old messages; the age limit will default to 1 week. The user will be able to fine-tune this value but not remove the behavior. The range that is being proposed is 1 to 3 weeks,

 © Research In Motion Limited, 1997

where a target number of messages stored at any one time is between 25 to 50 messages, depending on size.

The normal user would normally read and delete received messages. This step alone would help avoid the need for purging information. The Leapfrog user will also have the option of moving a received message in the Saved area. However, the message management strategy described above would apply to all message areas except the saved message area. The saved message area either would wait 'twice' as long before purging or would have a maximum saved message limit.

The following is a brief description of all the message areas on the Leapfrog:

- **INBOX**: The message view for incoming messages both read and unread. Messages that are unread have a special indicator that is removed once a message is read. Message can be moved from this view to the 'Saved' view if necessary.

- **OUTBOX**: The message view for outgoing messages. These have indicators to show whether they have been sent successfully, whether they are pending or confirmed.

- **SAVED**: The message view for important information to be saved by the user. These messages could be reminders, messages to be sent at a later time or daily reminder messages.

There will be a method for automatically deleting messages from the **saved** message area, based on a combination of message age and the number of messages in the area. This may or may not be a configuration item. (The user does not want to be overloaded with configuration options.)

Messages can be created on the Leapfrog and sent through the Gateway to a variety of destination services. A message can be created in several ways:

- **Self-Authored**: The user creates a message from scratch and either uses an address and addressing method from the address book, or creates an address dynamically for a one-time transfer. The user can address the message to multiple destination addresses or include addresses on a cc: list.
- **Reply**: The user can reply to a received message, which will automatically use the sender's address for the destination. The default address can be changed or added to if the user desires, or a cc: list can also be added to the message reply. The original message contents will be sent with the message. If a message comes in from another pager it should go back out the same path, i.e. directly to the pager.
- **Forward**: The user can forward a message which is similar to reply, except the original address is not kept.

After the message is constructed for transmission, the user normally will immediately issue a 'send' command to cause transmission to start, but other options exist for the user. The user can also save the message for later transmission. The user also has the ability to resend the message later, in case of a transmission failure or packet return.

During the transmission of a message, most users are concerned with the progress of the transmission. This concern will be satisfied by adding an indicator to the message, similar to the Inter@ctive Pager. The user will be able to check the Outbox and see changes to the message as it reaches the network. The Outbox message list will show several status indicators, these include:

- **Message Pending**: The message is still pending to be transmitted, either because other messages are queued before it or because there is currently no coverage.

- **Message Sending**: The message is currently being transmitted. It has been given to the radio component and the application is waiting for the 'sent-to-network' indicator.

 © Research In Motion Limited, 1997

- **Message Sent**: The message has been sent to the network and for 99.9% of messages, it will make it to the Gateway. If the Gateway is down or congested the packet will get returned and this will be shown.

- **Message Returned**: A transmitted message has been returned by the network. This is because either the FST was down, a network congestion error occurred or the mobile is not activated.

- **Message Mailboxed**: Normally used when sending peer-to-peer messages. This indicates that the mobile being sent to is currently out of coverage and the page has been mailboxed.

- **Message Confirmed**: If supported by the gateway, a confirmation message has been received. This could indicate the message has been submitted to the destination media (Fax, One-Way Pager or SMTP Internet Mail), or that the Internet mail message has reached its destination.

If the Gateway does not support the delivery confirmation then the final message display type will not be supported.

Note : It should be noted that the recent change by RAM to support PIN numbers for peer-to-peer transmission must also be implemented in Leapfrog. It should also be noted that peer-to-peer messages may also be followed by a range of status message from the Gateway as to the progress of message delivery. These indications should be handled as they are today in the Inter@ctive Pager.

NOTE : Would it be possible instead of showing the time in ASCII text to use a icon with the hands of a clock on them? (🕒p 3:00 pm, 🕓a 4:00 am, 🕔p 5:00 pm, 🕘a 9:00 am, and 🕚a 11:00 am) Interesting idea that I received from a Co-Op interview; it does save some screen space on every line.!?

NOTE: VERY BIG QUESTION : Does the Extended HP-ID 4 Specification apply when using RTP??

 © Research In Motion Limited, 1997

Address Book Sub-System

The address book will be similar to the Inter@ctive Pager today. The user will be able to added, change and delete address book entries, which are used primarily for authoring and sending mail to other people.

Address Book	Description
Address Book API	The addresses book with export functions for all other sub-systems to call, mainly for compose and attachment
Recent List	Most initial address book lists show a cache or most frequently accesses addresses, cache size is configurable
Full List	The normal view will be a complete 'sorted' list of all addresses, also called the full list
Search On View	Within each view of the address book (recent or full list), the user can search or narrow the list by setting up a search criteria to create a subset of the current list
Select Process	The selection process involves picking a name, and then picking a 'method' for message delivery. When there is only one method, the later step will be skipped
Add Addresses and User Defined Fields	The user will be able to addresses and personal fields to the address entry; limits on new fields will exist
Add Addresses Programmatically, including Fields (keys, etc)	Other programs will be able to addresses and special fields programmatically; encryption keys may be necessary if we go to public keys
Edit Address Data	The user will be able to view and edit address data using the address book sub-system
Sort Views by first name	The user will be able to sort the Recent and Full view by a one of three methods, including : first name, last name and company name
Customize	The user can customize the address book, primarily in the number of cached items in the recent list, the new fields added to each address book entry and default view

The address book is a database of names that can be used for general reference or for construction of mail messages. The current list of the default fields supported in the address book:

- **Name**: Both the first and last name of the person being referenced.
- **E-Mail**: E-Mail address for sending a message via SMTP to the Internet.
- **Fax**: Fax number for sending a message to a destination Fax modem.
- **Phone**: A phone number for sending a message, via text-to-voice conversation to a destination phone number. This is normally just for reference and holding contact information.
- **One-Way**: A one-way pager PIN number for sending a message to a one-way pager
- **Two-Way**: A Leapfrog Pager MAN/PIN or Inter@ctive Pager MAN/PIN that will be used for sending a message through the gateway to another Pager. The Gateway will be responsible for converting from RTP to HP-ID 4 for the delivery and for PIN resolution to a MAN.
- **Address**: the actual physical address of the person referenced in this address book entry.
- **Misc.**: Some number of remaining elements, perhaps a Note, Cell Phone number and Home Phone Number. The final selection of these has not been determined at this time.

The overall functionality within the address book sub-system will be similar to that of the Inter@ctive Pager. The user will be able to create, modify and delete address book entries up to 750 entries in total. When the user creates a message they will be taken directly to the address book to select a destination for the message. The user must enter at least one of the routing fields when creating an address book entry.

 © Research In Motion Limited, 1997

Phrase Book Sub-System

The phrase book is similar to the message book within the Inter@ctive Pager. It contain frequently used phrases that can be inserted into mail messages to save time in typing messages.

Phrase Book	Description
Phrase Book API	The phrase book exports an set of API calls so that the other sub-systems can make use of the phrase book
Categories	The phrase book will place phrases into categories, the use has the ability to add categories ??????????????????
Transmit/Display Strings	Each phrase in the phrase book can be set up to display one string and transmit a different string
Editable/non-editable categories	Some categories can be setup to be non-editable; this protects some phrases from being changed or deleted
Phrase Completion	When in an edit field, like the message compose screen, the user can press a key to have a phrase completed automatically
Phrase Completion (multi-word)	The phrase completion method can also substitute multiple words for a single 'short cut', i.e. RIM converts to Research In Motion
Customize	The user can customize the phrase book by setting up the following parameters ???????

The Phrase Book is similar in concept to the Address Book, but contains a database of strings. These strings are used to speed up message entry by allowing the user to insert the strings into actual messages to be transmitted. These short-cut strings are sorted into categories, similar to those used on the Inter@ctive Pager. Some of these categories include:

- **Appreciation**: Appreciation, gratitude and thank-you based phrases messages and statements, which are normally used to open or close messages.
- **Closings**: A collection of strings that are considered closings or terminating messages in a mail message.
- **Greetings**: A collection of strings that are considered greetings or opening statements in a mail message.
- **Miscellaneous**: Any other strings that are added to the system for insertion into mail messages.

There will be other categories; a full list will be decided closer to completion of the Phrase Book sub-system. The user will be able to add, delete and edit categories as they can on the Inter@ctive Pager today.

The user will be able to create, modify and delete phrase book strings, up to 750 phrases in total. When creating a phrase book entry, the user will be able to type messages directly into the Phrase Book under a given category. Every string must be placed into a category by definition, there will be no way to enter a string without entering a category.

 © Research In Motion Limited, 1997

Additional Sub-Systems

These additional sub-systems are not expected by RAM, but RIM would like to include them into their service offering. This means that they may not be included in RAM software release, but they will be a part of the RIM software service offering.

Other applications	Description
Basic Calendar	Provides a month at a glance view with no provisions for notes, or events (scheduler functionality is **not** supported)
Alarm Clock	A basic alarm clock that can be set to go off at any time on any date. Uses one of the preset **Tunes**.

 © Research In Motion Limited, 1997

The Stack

The RIM stack must support both RIM and RAM's protocol requirements.

Transport	
MDP	RIM's internally defined protocol stack
RTP	RAM's protocol stack – replaced HPID4
Compression	
Encryption	Encryption with a key of 0 initially (RIM only)
Datagram API	The stack exports an set of API calls so that the applications above can access the network. **For future expansion.**

 © Research In Motion Limited, 1997

Desktop Support Software

To assist in the support and maintenance of the Leapfrog there will be a desktop tool for the configuration and management of the Leapfrog. This product should be similar or exactly what RAM has today, which means the Leapfrog software should be compatible with the software protocols.

Basic Desktop Support

To satisfy basic requirements for RAM and for low-end RIM Pager users a basic configuration program is required. Some of the functions in this piece of this software will go beyond the current software and will include:

- **Advanced Configuration**: Secure configuration for network operator level features; there may be very few of these for Leapfrog since it cannot be used as a normal modem.
- **Network Configuration**: Allows a service provider to set options like gateway address and encryption keys within the pager.
- **Basic User Configuration**: User based configuration control to allow the user the ability to set certain parameters and make working with the device easier.
- **Off-line Backup**: This new feature will allow the user to store all their address and phrase book information in an off-line location. This allows for maintaining a backup of information on the Leapfrog, or quickly setting up a new Leapfrog with custom address and phrase books.

The current plan is to restrict what parameters can be updated over the Mobitex network. Some parameters, password, backup and restore of address and phrase book will only work when the device is connected locally via the serial port of the desktop computer. Most Leapfrog users will not have a secondary radio modem to perform the updates over the air, and would want to update their message store locally via a serial modem.

Specific application parameters, like the Gateway address, can be updated over the Mobitex network. To perform this update, a security password will be set up for each of the three configuration levels. This password will be maintained between the Leapfrog and the desktop configuration package that will be managed by different parties, i.e. user, service provider and network operator. This is similar to the remote configuration packet being used with the Inter@ctive Pager today.

Advanced Desktop Support

For those uses that are using a product that supports Microsoft's Messaging API (MAPI), they have the option of using Outreach at their desktop. Outreach is a mail redirector and backup/restore utility combined. It provides to the user a bi-directional path between their pager and their desktop. This allows mail, contact and schedule information to be sent to the user and for mail to be sent back from the user. In later stages contact and schedule information will also be sent back to the desktop with mail. The basic functional of Outreach includes:

- Setup and configuration of pager and environment for redirection, including initial download of selected personal and global addresses from MAPI message store.
- Ability to start redirection either through an idle period, like a screen save kickoff, or through a signal from the pager to the desktop. Redirection can be filtered by the user based on the user-id and other criteria like size and preferred/non-preferred addresses.
- A log file is kept of all message sent and received through Outreach, which can be reviewed later when the user comes back to the office.
- A desktop icon can be used by other employees in the office to contact the user while on the road. They can send messages and receive responses directly to the pager icon.
- A set of information can be redirected, including messages, address information and schedule information. With the release 1 application only message and addresses can be imported to their appropriate sub-systems, the calendar information will simply stay as a mail message.

 © Research In Motion Limited, 1997

Stage 2 Requirements

UI	Description
Complex Tables	Tables with advanced formatting options – supporting different fonts, bitmaps (?). Required for TK and some of the games
6 Line Mode	A display mode that supports a larger font.
ALT Indicator	An indicator displaying the state of the ALT key – this is more of a system design issue that should possibly be solved now.
Shift Indication	An indicator displaying the state of the Shift key – this is more of a system design issue that should possibly be solved now.
CAPS Indicator	An indicator displaying the state of CAPS lock – this is more of a system design issue that should possibly be solved now.
Password Edit Box (******)	A dialog box that handles password hiding (the *'s instead of the actual characters
Alignment of Multiple Edit Fields	
Justification of Edit	Left, Right, and Centered justification
Radio Buttons	A method of grouping multiple buttons, where only one can be selected.
Check Box	A box that can be marked as selected

Options	Description
Security	Addition of different security levels, causes the device to ask the user to enter a 4 digit Personal Identification Number (PIN)
Key download & protocol	Although the device had encryption, it used a key of zero; enables downloading of keys for both the service operator and the desktop user

Email	Description
Forms	Support for a portion of HTML 4.0 – unsupported formatting is either ignored or filtered by the host.
Scripting	A subset of the Visual Basic command set.
Message Formatting (CMIME)	**Move to V1???**
Free Form Addressing	Typing of a free form address on the To: line – requires prototyping before look and feel issues can be solved.

Address Book	Description
Sort on other....	Sorting on fields other then first name, last name or company
Import Addresses – view attachments	Support for address entries sent by attachment
Import addresses – restore	Restore addresses via the Backup/Restore utility – **MOVE TO V1**
Export Addresses – transmit attachments	Support to sending addresses via attachments
Export Addresses – backup	Backup addresses via the Backup/Restore utility – **MOVE TO V1**

 © Research In Motion Limited, 1997

Address groups	Local grouping of addresses on the pager.
Search for remote address info	
One Time addresses	Ability to use an address without adding it to the address book on a permanent basis
Copy/Paste of address objects	Copy and paste addresses as an object (into groups, messages, etc)
Collapse/Expand of address lists	Collapse sections (the 'D's) of the list when there are more then a set number of them. May not be required since same essential functionality is provided by using the search capabilities of the database to limit the view.

Phrase Book	Description
Nested Categories	Categories can contain sub-categories
Invisible categories	Categories not visible to the user
Invisible phrases	Phrases not visible to the user
Non-editable phrases	Read-only phrases
Auto-Correct	A limited dictionary that can be used to convert typing mistakes such as *taht* to *that*. Can also be used as a shortcut to phrase expansion: *wrt* expands to *with respect to*
Intelligence (auto-add phrases)	Applications can build up the phrasebook dictionary automatically depending on the text the user enters in sent messages (or from other sources)

TK	Description
Calendar	A full featured calendar supporting event indication: different fonts for dates with events, tasks, notes, etc. Supports direct links to Task List, Scheduler, Notepad, and Journal below.
Task List	
Scheduler	
Note Pad	
Journal	
Outreach	Full support for Outreach remote scheduling

Games And Other Apps	Description
Chess	Two player
Battleship	Two player
Connect 4	Two player
Calculator	Two player

 © Research In Motion Limited, 1997

Stage 3 Requirements (or later)

Public Key Encryption is the currently the only foreseen development in stage three. Adding support for public key encryption requires changes to the stack, the messaging sub-system, and possibly the address book as well.

Further stage 3 requirements will be added as the product matures, and feedback about version 1 (and possibly version 2 as well) is returned.

 © Research In Motion Limited, 1997

© Research In Motion

Design Ideas – Outreach Product

Introduction

After some struggles with what Outreach is we may be close to understanding how it should work. The current design ideas for the product seem to indicate that Outreach should be divided into three related products, or three faces of the same product, these three products include:

1. An always present background monitor that sits in the sys-tray (right side of toolbar) and performs low-level activities on behalf of the 'Outreach paradigm'.
2. A log viewer and activity monitor that allows the user to create a configuration profile and view activity that has occurred with Outreach over a specified period of time.
3. A screen saver interface module that allows Outreach to appear as part of the user's screen saver. This screen saver sits in the screen saver 'activity chain' and does not disrupt the existing screen saver.

What has been difficult to finalize is the relationship between these three faces of Outreach, presented in the remaining section is one possible implementation of this relationship.

System Tray Applet

The System Tray applet is a program with virtually no interface. It is run as the main or central Icon for the Outreach product. During installation the user has the option of placing it both in the Startup folder and on the desktop too. When the program launches it goes directly to the sys-tray, unless by some chance it is detected that configuration has **never** been performed.

Log Viewer

The Log Viewer and Configuration program is either run from the Sys-tray icon or from the Outreach folder. During the installation process the user will have the choice to set up some configuration components before actually running the program. The log viewer and configuration manager has two purposes, it can show all the logs for a given area, and it can change configuration for a given area.

Screen Saver Applet

The screen saver applet is used to connect into the existing screen saver for activation when the user's screen saver kicks in. It is used primarily to reach the user when they are away from their desk. The owner of the Pager can set up the screen saver applet to hook into the existing screen saver and even give the applet a special message to display when it appears. Someone who types into the pager will reach the user and be able to track them down in time.

Startup Stage

Ideally when the computer is started some form of Outreach icon will appear in the system tray to indicate Outreach is running. If Outreach has never been run before the main Outreach configuration window will appear. (As time permits we should perform a series of Installation Wizards to help get the initial configuration correct for the product to avoid the potential problem of not have an initial configuration).

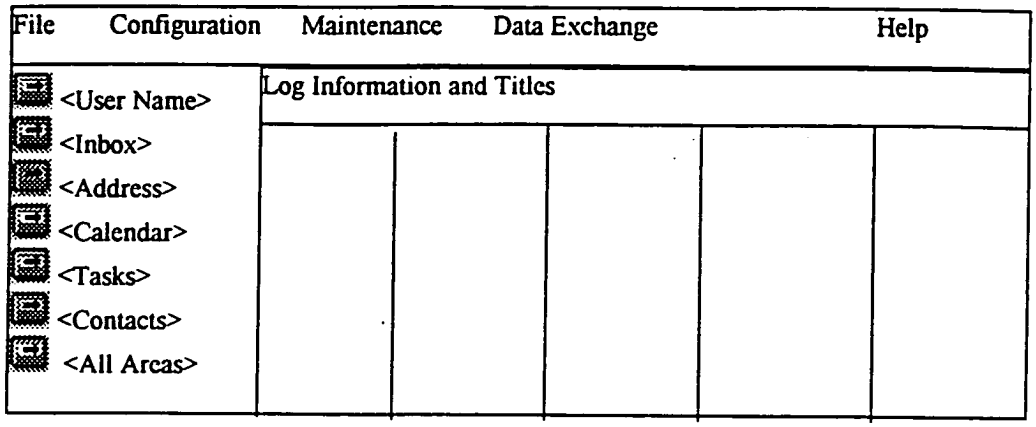
After loading a user can right click on the Icon to bring up the following menu shown in the table. When the screen saver hook is enabled the option for enabling is grayed out. When the screen saver hook is disabled the disable option is grayed out. The Screen Saver Configure option will cause the screen saver Pager UI to appear and the user can set up the options for this interface.

Double clicking on the Icon will bring up the Log View and Configuration screen shown later. A single click will perform a ??

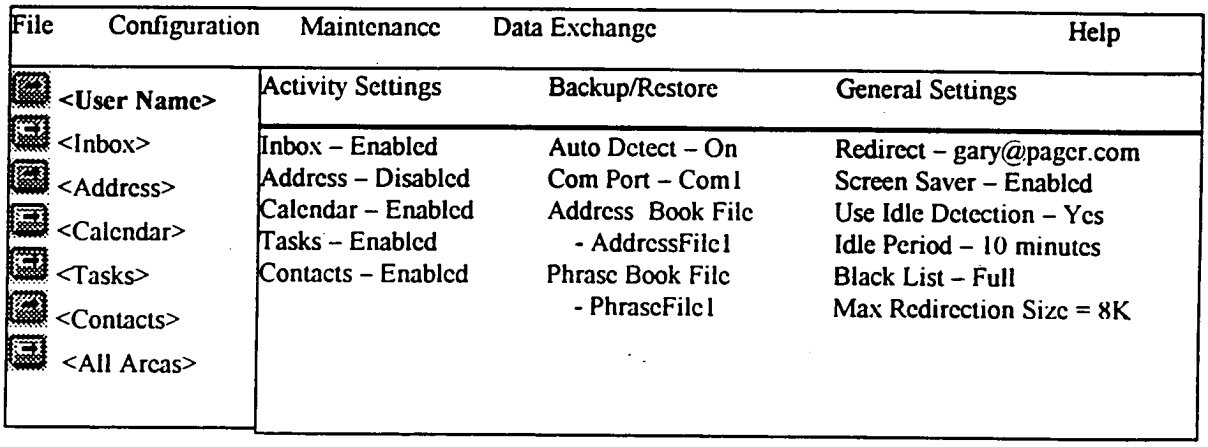
© Research In Motion

Outreach Viewer

This main Outreach screen is used for logging and configuration of the Outreach product. It looks something like the following:



Initially the screen is blank until the user selects an area for viewing or manipulation. If <User Name> is selected this replaces the Log Information area with a summary of configured elements of the product. This includes black list summary information, screen saver settings, redirection enable/disabled settings and target redirection mailbox name. When the <User Name> Icon is selected the following is a sample screen that will be seen by the user:



Note : Using Idle Detection would override the use of the screen saver to determine when to redirection information. If both features were enabled the idle period timer would be used.

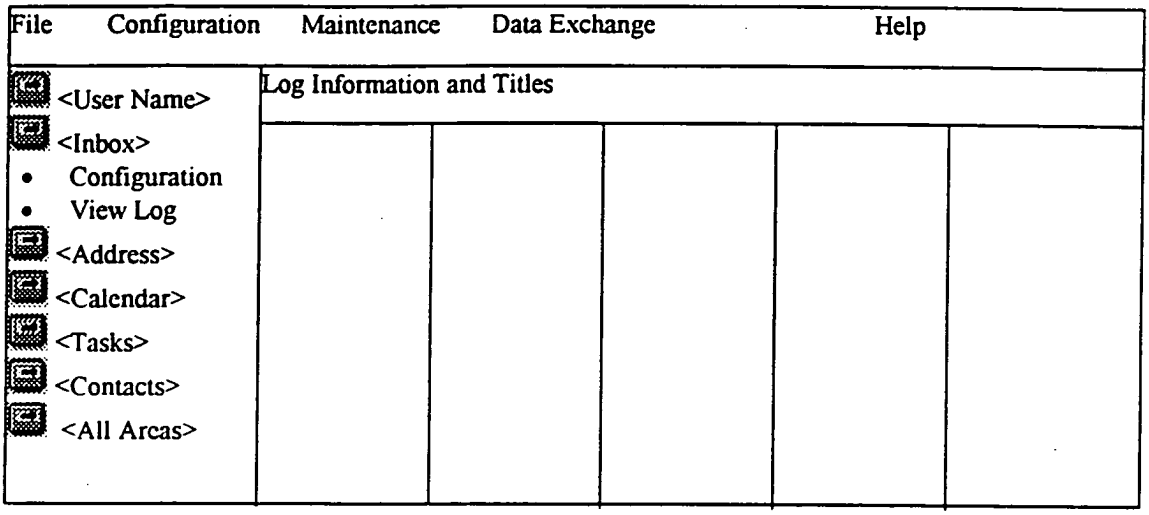
There will be other configuration values, but this is the most important list that will be highlighted in this document. With **User Name** selected the Configuration menu item will have the following choices: *Redirection Address, Screen Saver Configuration, Idle Period Configuration* and *Filtering Configuration.* Naturally in each of the configuration areas will be some user interface screen to configuration the different areas.

Under the Maintenance menu item will be choices like: *Clear Log, Save Log and Clear, Autosave Log Daily, AutoSave Log Daily* and *AutoDelete Log Daily.* There could be many of these choice so there may be a better way to do all these choices. It may be possible to put some of these into the File menu item to help organize the many choices.

Under the Data Exchange menu item will be choices like: *Edit Pager Address Book, Edit Pager Phrase*

Book and *Download Pager Addresses* and *Download Pager Phrases.*

The user can go through each of the other selections and perform configuration and log viewing When the Inbox, or any of the remaining items are selected there are two possible ways it should behave. The first way is for the Inbox to expand and open up to show two different choices:



An **alternative** of this is if the user selects Inbox they always get the Log Information, but they can also go to the Configuration menu item and select the following: *Black List Configuration*, *Message Size Configuration*, *Enable Redirection* and *Disable Redirection.*

Under the other items are different configuration items, for example under Address may be a *Enable Address Redirection* and *Disable Address Redirection.* Under the Maintenance menu item would be a choice like: *Update Pager Addresses*, or even more difficult *Import Pager Addresses.* The second item is new but it would allow addresses entered on the Pager to be placed back into the Outlook Personal Address Book.

© Research in Motion

OUTREACH SOFTWARE DESIGN

Version 1.0
December 3, 1997

Product: Inter@ctive & Leapfrog Pagers
By: Gary Mousseau

Proprietary and Confidential
To
Research In Motion Limited

© Research in Motion

Table Of Contents

Design Ideas – Outreach Product 3

Introduction 3

Main System Process 3
MAPI Redirector 3
Screen Saver Applet 3

Data Considerations 4

Startup Stage 4

The Automatic Touch 4

The Inbox 7
The Address Book 8
The Calendar 9
The Contact Folder 10

Main Menu Options 11

Design Ideas 12

© Research in Motion

Design Ideas – Outreach Product

Introduction

The functional definition of what Outreach has been very challenging. The current design ideas for the product indicate that Outreach should be divided into three main products, a main Outreach window, the hidden MAPI redirector windows and a Always In Reach (AIR) screen saver process. The general functionality of these three components includes:

1. The main Outreach window also includes an always present background monitor. The background monitor presents a sys-tray icon (shown on the right side of toolbar), similar to the Palm Pilot, REX and other products. By selecting the sys-tray icon the main Outreach screen can be easily reached and activated. The main Outreach program also includes a download and upload utility program.
2. A hidden MAPI background redirection process, that for stage 1 of Outreach lives on the same machine as all the other Outreach processes. It reads information from the MAPI message store to determine its behaviour.
3. A screen saver interface module that allows Outreach to appear as part of the user's screen saver. This screen saver sits in the screen saver 'activity chain' and does not disrupt the existing screen saver. It is capable of a simple send of an immediate page to the user and is more a marketing trick.

One of the key goals of the Outreach product is to be a simple to use, intelligent program that is capable of auto-detecting the presence of Leapfrog or the Inter@ctive Pager (depending on the version of Outreach) and performing the expected and configured action for the user. It is expected that the Outreach product will initially be used with the Inter@ctive Pager but be expanded and further developed to work with the Leapfrog Pager.

Main System Process

The System Tray applet is the main program seen by the user and is also tied to the system tray (tool bar). It is most commonly used to perform a **traditional** synchronization with information at the user's desktop, configure the redirection process and view any log information that might exist. During installation the user has the option of placing it both in the Startup folder and on the desktop background screen. When the program launches it goes directly to the sys-tray, there will be no option here, this is because it also indicates that the redirection process is in placed.

MAPI Redirector

The MAPI Redirector is a background process with no main screen. It is always running with the sys-tray program is running and is stopped when the sys-tray icon is stopped. This tracking behaviour is used to ensure that the MAPI Redirector is not left in the system by accident. Normally the activity performed by the MAPI Redirector would simply be integrated into the Main System process, but this would be a bad design (see Internal System Architecture section for more details).

Screen Saver Applet

The screen saver applet is used to connect into the existing screen saver for activation when the user's screen saver kicks in. It is used primarily to reach the user when they are away from their desk. The owner of the Pager can set up the screen saver applet to hook into the existing screen saver and even give the applet a special message to display when it appears. This message can be typed in by the user, come from the current scheduled activity or be sent from the pager to be displayed on the screen saver when on the road.

© Research in Motion

Data Considerations

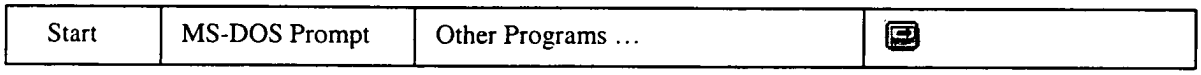
It has become clear from looking at other software packages, and reviewing the type and nature of the data RIM is dealing with, that there are three types of data RIM's Pagers could be working with. These types include:

1. Static Data : Data that only changes because the user changes it themselves, it is unlikely that anyone but the user will change this type of data.
2. Dynamic Data : Data that is changing independently of the user. This can include messages, alarms and any notifications that could appear on the user's desktop.
3. Mixed Data : Data that could cause dynamic events to occur that are independent of the user, this same data may also have been setup by the user.

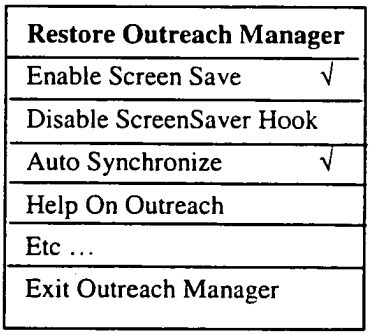
With these three types of data being present in the RIM Pager design it is clear that different models for managing these data types is required. Based on this reality the necessity of using a partner like Puma may be required to ensure that RIM's static data models are well supported.

Startup Stage

Ideally when the computer is started some form of Outreach icon will appear in the system tray to indicate Outreach is running; a simple example is shown below:



If the main Outreach process had never been run before, then when it is started the main screen appears first. Otherwise the sys-tray Icon will the first thing the user sees of Outreach. By clicking with the right mouse on the sys-tray icon the following menu appears.



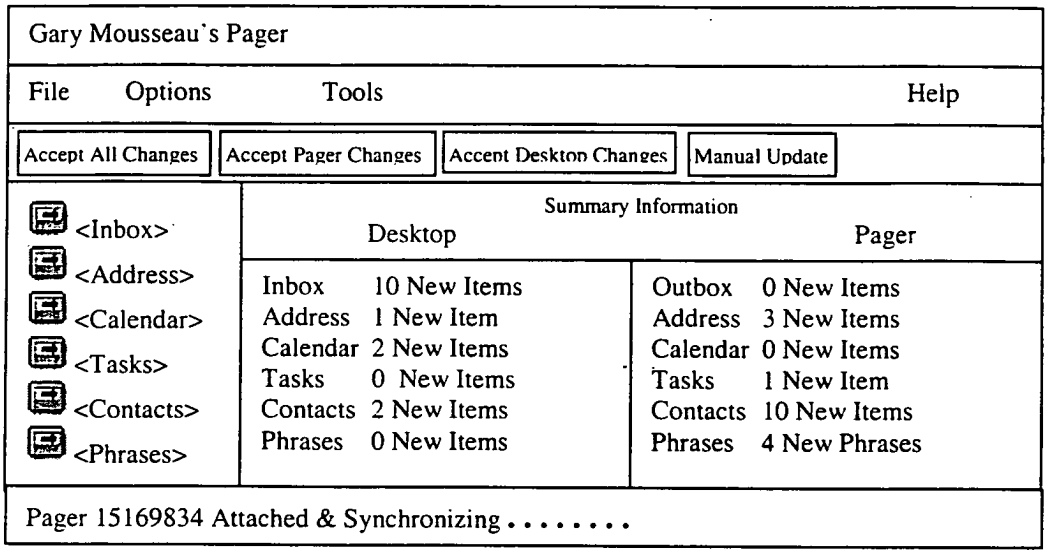
The first menu is bolded because by convention this is the action if a double click is preformed on the Icon. The screen saver hook can be enabled or disabled directly from this screen. The Auto Pager Detect will allow the software to automatically detect the presence of a Pager device that has been plugged in, for instant communication establishment. Similarly, the Auto Synchronize indicates that the user wants synchronization to take place every time the device is placed into its serial cable. **When there is no serial cable is available Outreach will do whatever it can over the wireless link.**

The Automatic Touch

The default setting of the Outreach product is to easily and transparently detect a connection to the Pager and automatically perform several steps. This also involves changes to the device in order to be watching the serial link for an Outreach protocol handshake. With most other devices the user must enter a special mode, like the PageWriter 2000 for example, but with the RIM Pager the device automatically enters a data exchange mode.

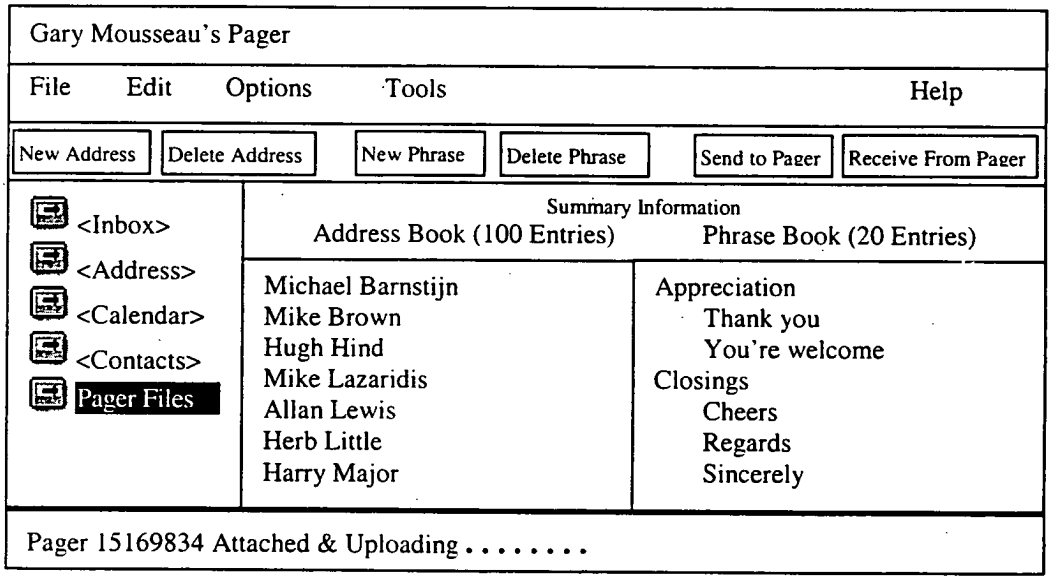
Once plugged in, and the auto pager detect flag is turned on, the small sys-tray icon maximizes to full size and indicates the device is plugged in and shows its MAN number. If the auto synchronize is turned on Outreach will perform an automatic download, followed by an automatically comparison of the records. This will be a Leapfrog functionality since the Inter@ctive Pager will not be changed to support this conversation.

When working with Leapfrog, the maximized Outreach screen will present as much information as necessary to give the user a simple one-button type of decision process. The following screen is a simple representation of a possible screen layout for the auto-synchronization.



For the Leapfrog Pager, once the pager is plugged in and the sys-tray icon expands, the user can quickly see a summary of what is happened since the last time they plugged in, as shown above. This advanced method of synchronization may only be possible with the Leapfrog Pager. The row of choices through the middle (Accept ...) would be Icons representing a toolbar, supporting full tool tips.

The Interactive Pager has a much more static information exchange, which would make the method shown above almost impossible to achieve. As a stage one the backup/restore functionality may look something closer to the following:



In this example the user could simply select an item via a single click, or double click on the item to edit it. Once an item is selected it could be deleted, or a new item could be added in that area. During this initial stage of download from the page, the item called Pager Files is highlighted. This is the tab for getting to the currently, or last saved pager files. The toolbar provides a quick short cut for send information to the

pager or retrieving information from the pager, in the real implementation items like open and save would also be present.

In both cases once the pager was removed the window would minimize automatically back to the sys-tray. The goal is to make interacting with either pager easy and simple.

© Research in Motion

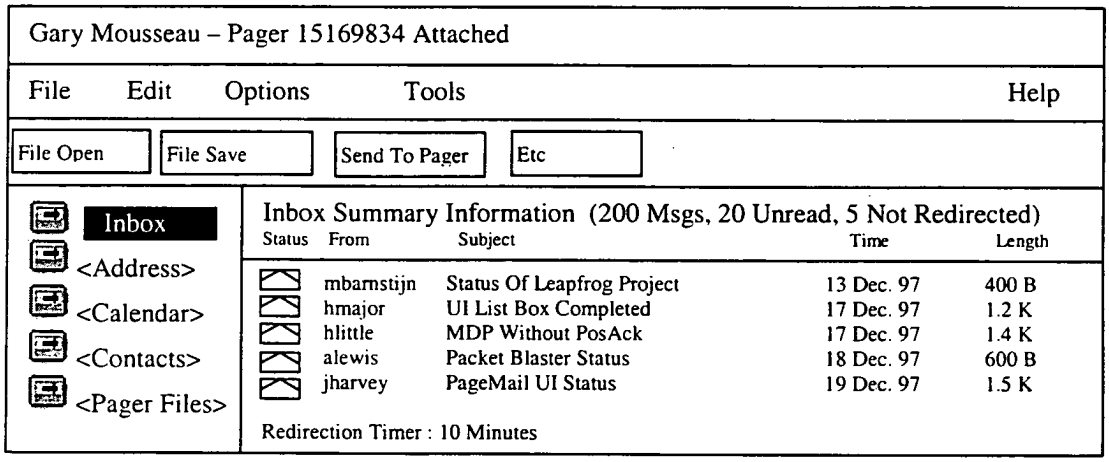
Outreach Viewer

After plugging in the pager and getting the screens already shown, the user can move into other activities with the main Outreach viewer. In general terms the program acts as a message manager, and if necessary the log file can be accessed to see what has occurred while the user was redirecting messages. Each tab on the left side of the main window represents an information area the user can work with. For the initial Outreach, working with the Inter@ctive Pager, the Inbox, Tasks and Calendar left menu items will not immediately bring up values in the rights most windows. (Examples are shown below).

By right clicking on each of the left-side icons the user will be given a short cut to configuration and viewing the log information for that section. In subsequent stages, for implementation of Outreach for Leapfrog, additional behaviour will be added to the Inbox, Calendar, Tasks and Journal areas.

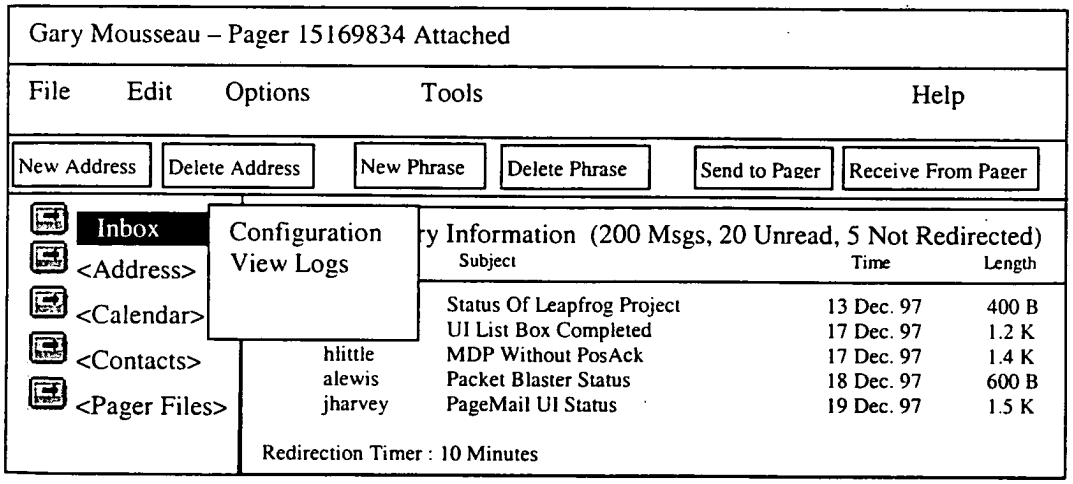
The Inbox

When the user selects the Inbox tab, when working with the Inter@ctive Pager, the following behaviour will be expected:



The goal of this screen is to inform the user what has **not** been redirected to the pager in a quick glance. Then they can decided to manually force a redirection of any of these messages if they desire – for on the road viewing, or if the user has had their secretary come to their desk to redirect it for them. In the case of e-mail and the Inter@ctive Pager these must **always** be sent wirelessly to the device not over the serial port.

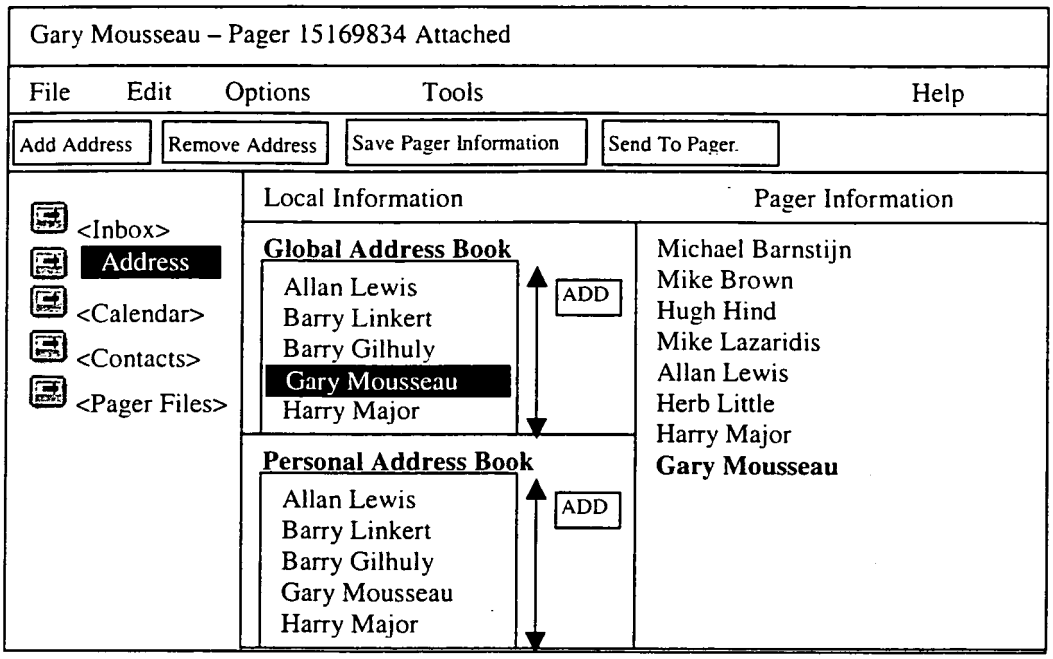
The tool bar may adjust when the Inbox, Address, Calendar, Contacts or Pager Files items are selected (depending on whether this is 'reasonable' Window's behaviour). On a right click the following options appear:



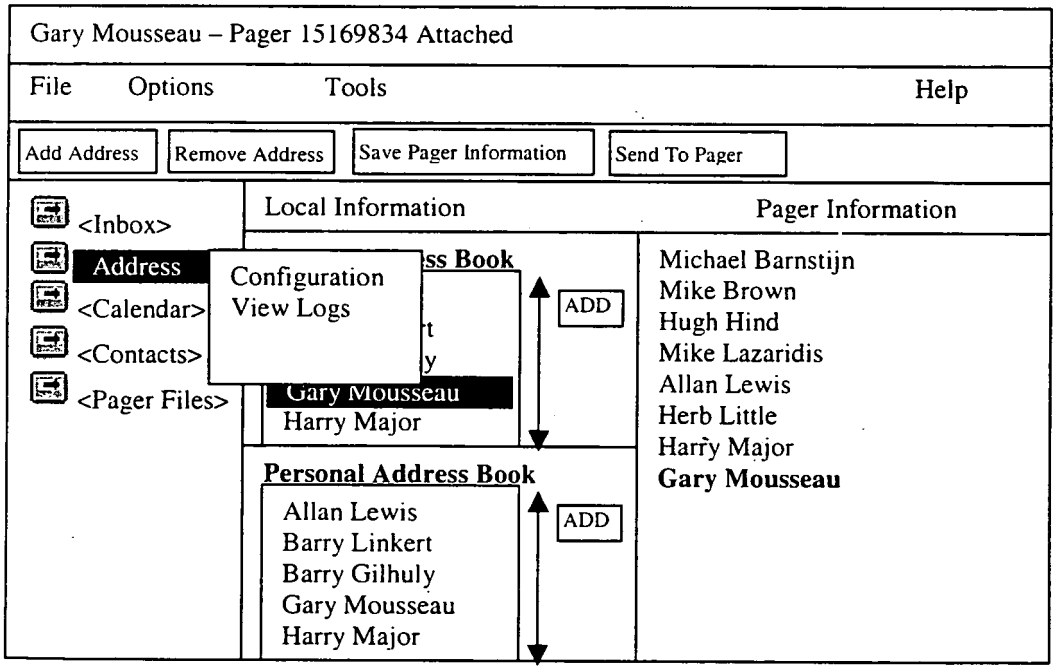
© Research in Motion

The Address Book

When the user selects the Inbox tab, when working with the Inter@ctive Pager, the following behaviour will be expected:



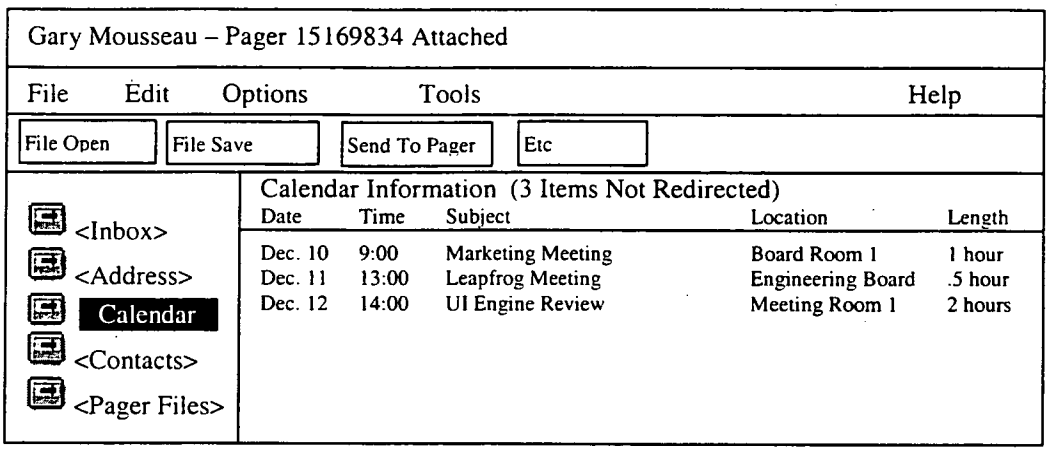
When the user moves to the address item and clicks they get a quick view of the global and personal address book of the user. If the user moves to Gary Mousseau and clicks on ADD the name is added to the current items in the Pager Information area. Each name that is added to the 'Pager' files is left **bolded** until such time as the user goes back to <Pager Files> and performs a Send to Pager command. The Send to Pager command will remove the **bolded** entries since they will no longer be unsynchronized. If the Pager is **not** attached, these items are sent via **mail** to the pager – see last part of this section for details. If the user tries to exit with items that have not been downloaded a warning is given.



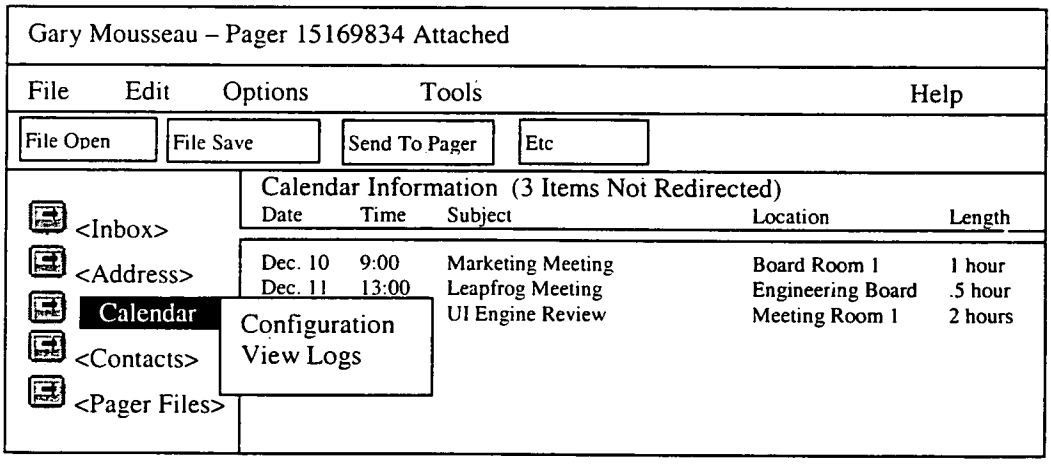
When the address book gets sent via mail to the Inter@ctive Pager it follows a specific format that allows the message to be imported as an attachment. When working with the Leapfrog the format will change but attachment concept is the same.

The Calendar

The Inter@ctive Pager and Leapfrog Pager are very different in this area. This is due to the fact that in time the Leapfrog product will have a full Calendar and Timekeeper application. However with the Inter@ctive Pager all Calendar events are converted to **mail** messages and shipped to the pager. To help facilitate this when the user selects the Calendar tab the following is expected:



Normally the user is manually entering schedule events into Outlook when they are sitting at their desk. Since the redirector has not taken effect, the user can either wait until the normal redirector sends each event in the scheduler to the pager when it kicks off, or the user can force the schedule events to be mail to the pager immediately. In this screen the user simply has to select an items and issue the Send To Pager command.

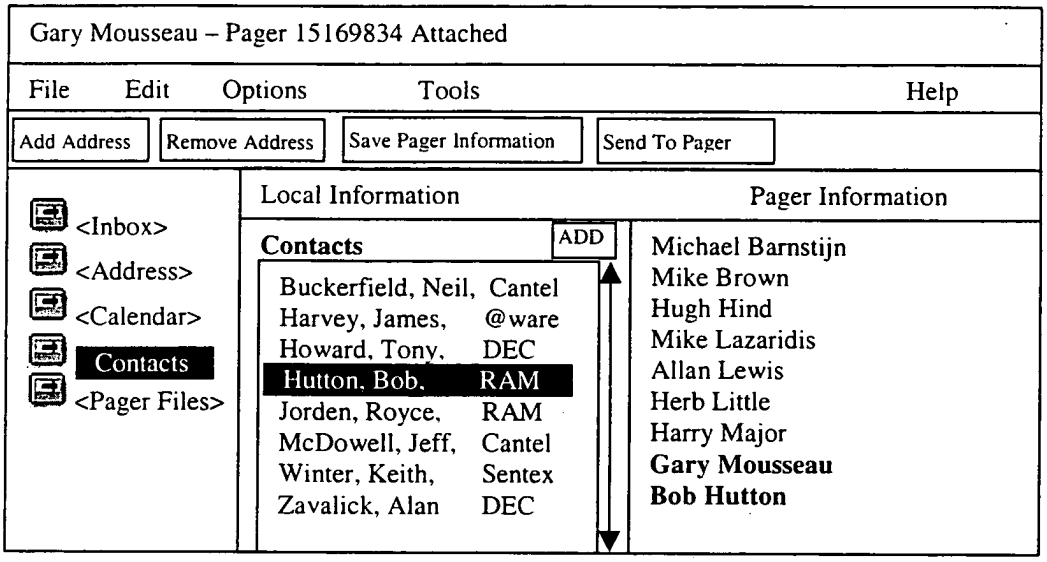


When the user Right clicks on the calendar item they are given short cuts to the areas of configuration and viewing the logs.

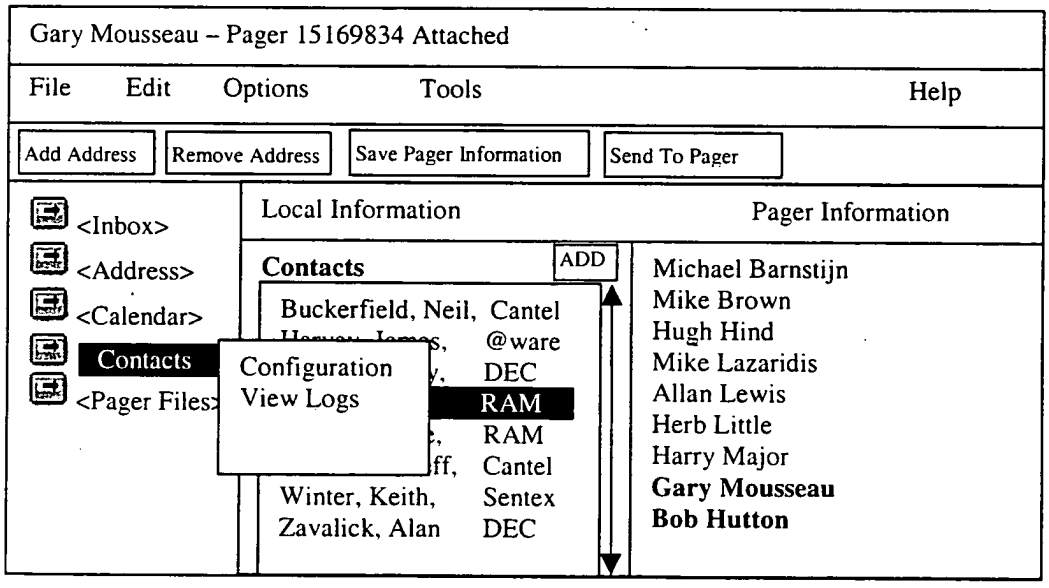
© Research in Motion

The Contact Folder

The Contact Folder contains names and addresses in Outlook that have been entered for contact management reasons. In the Inter@ctive Pager and Leapfrog Pager contacts and the Address Book are merged into one area. Just as when working with the address book, the user can choose to 'Send Contact Information to the Pager. If the pager is not attached the information is mailed in a way that is compatible with attachments for the Inter@ctive Pager. In this way the user only needs to go one place to get both addresses for E-Mail interaction and contact information for voice calls. When the user selects the Contact tab, when working with the Inter@ctive Pager, the following behaviour will be expected:



Just like in the address book, when the user moves to a contact item and clicks they get a quick view of all contact in alphabetical order from the contact folder. If the user moves to Hutton, Bob from RAM, and clicks on ADD, the name is added to the current items in the Pager Information area. Each name that is added to the 'Pager' files is left **bolded** until such time as the user goes back to <Pager Files> and performs a Send to Pager command. In the example both Gary Mousseau and Bob Hutton are new so they are bolder waiting to be sent to the pager. The Send to Pager command will remove the **bolded** entries since they will no longer be unsynchronized. If the Pager is **not** attached, these items are sent via **mail** to the pager – see last part of this section for details. If the user tries to exit with items that have not been downloaded a warning is given.

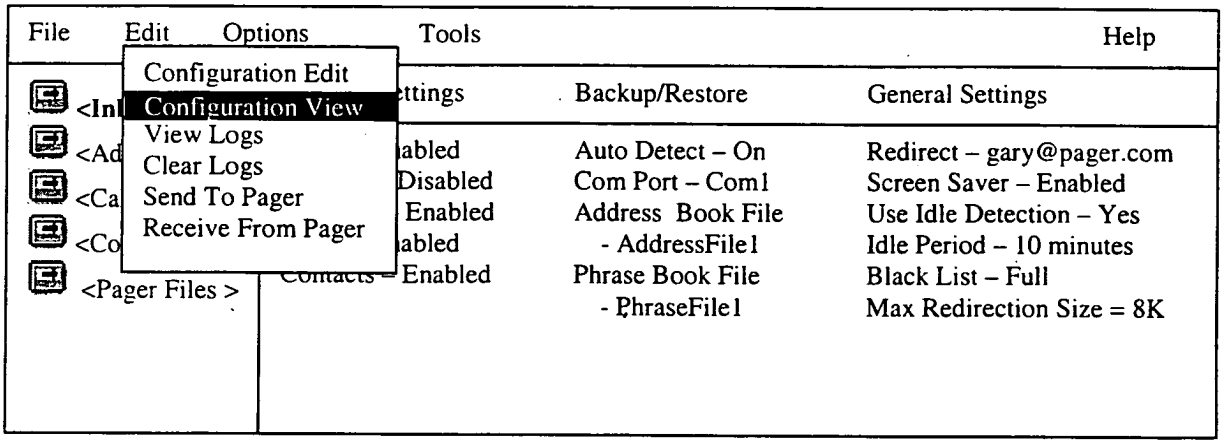


Main Menu Options

The following main menu choices under the 'File', 'Edit', 'Options' and 'Tools' menus are not final choices, but recommendations for satisfying the current requirements.

Under the 'File' menu will be choices like 'New', 'Open', 'Save' and 'Save As'. These choices will be mostly used for Opening and Saving the pager specific files. Under the 'Edit' menu will be choices like 'Copy', 'Cut', 'Paste', 'Delete Entry', 'etc'. These choices will be primarily focused on working with Pager specific files when adding and manipulating record entries in the phrase and address books. Under Options are the choices shown below. Choices like 'Send To Pager' and 'Receive From Pager' will be grayed out in most cases, except when <Pager Files> is selected.

When the user selects the Options menu item, they can view the current configuration settings or edit the current settings. If the user were to select Configuration View they would get a summary of all the configured information. This includes black list configured setting, idle detection period, screen saver settings, redirection enable/disabled settings and target redirection mailbox name.



Note : Using Idle Detection would override the use of the screen saver to determine when to redirection information. If both features were enabled the idle period timer would be used.

There may be other configuration values, but this is the most important list that will be highlighted in this document.

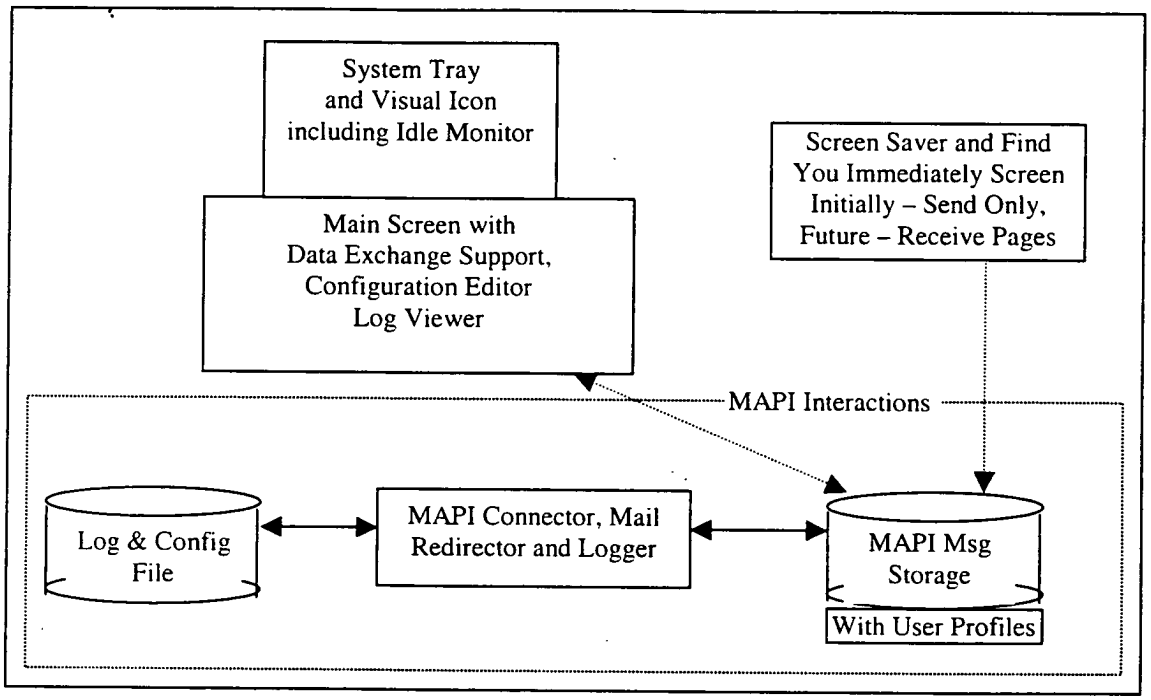
In the short term advanced features like 'auto log file' maintenance can be ignored. We have lost enough time due to poor specification and the creation of PageMail that we must not focus on the essentials with Outreach stage 1.

Design Ideas

This section presents some internal design ideas for Outreach, while considering the effects and advantages for the PageMail product. Outreach is composed of three components, the main Outreach task that has a direct hook into the MAPI sub-system and connects to the sys-tray of Windows. The second component connects to the screen saver and will initially allow a user to send a pager to find the user. The third component is the direct MAPI redirection code. This last components runs as a hidden application that will be started and stopped via the Main Outreach screen. This last component could be part of the Outreach main program, except when the PageMail product is considered.

To make the development easy, and to eliminate most inter-process communication, the MAPI interface method was going to be the underlying information exchange mechanism. For example the configuration of services and features should be down to a private MAPI folder that can be read by the main Outreach screen or from the Screen Saver code. The screen saver initially only has to write a message to the MAPI sub-system, and someday be able to monitor and watch for feedback or returned messages from the Pager. Similarly when the screen saver kicks off, or when the keystroke idle monitor, detects activation trigger it will make a change to the MAPI private folder. This change will tell the main Redirection code to being redirecting information.

In time it should be possible for the main Outreach screen to start working with the PageMail server. This way the user would be configuring different behaviours and the trigger message will then go to PageMail, which would stop messages being flooded to the pager when the user is sitting at their desk.



In Stage 1 the Sys-Tray Icon with the main Outreach screen is the central desktop focus and connects to the system tray or toolbar of Win 95 and Win NT. Most users will operationally use the Sys-Tray Icon to get at the main Outreach screen. This component should also, in time, monitor keystroke activity and when configured tell the redirection code that a message redirection trigger has kicked off.

In stage 2 the redirection portion of the code would exist on another machine. The user's desktop machine is sending signals, via MAPI, when redirection should take place. If log information is kept in this case it would go via MAPI to PageMail to get this information. Similarly when making configuration changes, the information is saved in a Private MAPI folder in the message store.

RIM CONFIDENTIAL DOCUMENT
Classification: COMPANY SECRET

Document Title:	LeapFrog Requirements Specification
Document Revision:	Version 1.0
Document Date:	8 December, 1997
Document Reference Number:	ENG-LEAPSPEC.DOC

This document is issued for signoff to:

Recipient Name: Mike Lazaridis
Recipient Company: Research In Motion
Sign Off Signature:

Recipient Name: David Werezak
Recipient Company: Research In Motion
Sign Off Signature:

Recipient Name: Michael Barnstijn
Recipient Company: Research In Motion
Sign Off Signature:

Recipient Name: Gary Mousseau
Recipient Company: Research In Motion
Sign Off Signature:

Signoff of this document indicates agreement of the specified version 1.0 functionality. Once signoff occurs there is a formal method for adjusting the functional defined in this document. By using an Engineering Change Request (ECR) it is possible to adjust and change the specified functionality in this document.

© RESEARCH IN MOTION LTD. 1997

Proprietary and Confidential

Table Of Contents

Introduction 1
Version Summary 1
Technical Summary 2
General Requirements Overview 3
Stage 1 Requirements 4
User Interface (UI) Sub-System 4
Options Sub-System 6
E-Mail Sub-System 8
Address Book Sub-System 11
Phrase Book Sub-System 13
Additional Sub-Systems 15
Desktop Support Software 16
Basic Desktop Support 16
Advanced Desktop Support 16
Stage 2 Requirements 17
Stage 3 Requirements (or later) 19

 ©©} Research In Motion Limited, 1997

Introduction

The decision to create the PageMail Server has left the Leapfrog's application software deadlines extremely tight. In an effort to reduce risk of failure, and risks of producing the wrong software, the Leapfrog group has produced this document in an attempt to describe the desired functionality of the Leapfrog application. It is hoped that the sign off process will encourage all the key individuals to review and carefully think about what the Leapfrog is, and what Leapfrog should be able to do.

The Leapfrog Requirements Specification is designed to be a detailed summary of what Leapfrog will be able to do in Version 1.0, and a terse summary of what Leapfrog can do in Version 2.0 and 3.0. This document's goal is to describe the functionality of the software, not present specifics on the user screens and visual presentation. There are several other documents that can be read if the reader wants to review exact screen presentations. Additionally, the scope of this document is to describe 'what' the user can do once they purchased a pager and are about to use it.

Version Summary

Given the short timelines for producing a functional device, and RAM's rigid timeframe for shipping a Beta level software product, the list of requirements has been broken up into a staged delivery. The first stage is specifically focused on RAM's requirements for the device, and those that are implementing it should keep the Inter@ctive Pager in mind. As such most of the advanced features will be held back until version 2, which is the first RIM version. It is not expected that RAM will receive a version 2, instead they will go through minor revisions of 1.0. RIM's version 2 will evolve into version 3 and eventually version 4. The expected delivery dates for the first 2 stages are as follows:

- Stage 1, Version 1.0 – Basic Device Functionality – At this stage, the leapfrog is an Inter@ctive Pager with everything the Inter@ctive pager does, and **very little more**, for example the RAMParts Transport Protocol (RTP) will be added.

Alpha Available	– Jan 15th
Beta Available	– Early February
Completed	– Late February

- Stage 1, Version 2.0 – First RIM Version with Advanced Functionality – Adding Compression, Encryption, Compressed MIME (CMIME) Encoding, OutReach Support, Cut/Copy/Paste, additional applications like a Calculator and other extended features to be determined from marketing feedback and internal requirements. Some back-end work to support the SWAP interface within the PageMail server will be complete in preparation for forms support in Stage 2.

Alpha Available	– Mar. 30th
Beta Available	– Apr. 15th
Completed	– Apr. 30th

- Stage 2 – Further Device Improvements to RIM Specific Application – Added Forms, Calendar, added games and extended features to be determined from marketing feedback and internal requirements. Public Key Encryption may be added at this stage depending on usage models.

Alpha Available	– Middle of 2nd Quarter
Beta Available	– End of 2nd Quarter
Completed	– End of 2nd Quarter

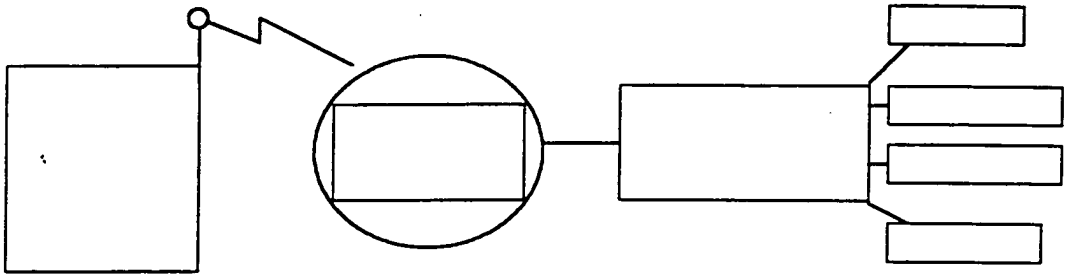
 ©©) Research In Motion Limited, 1997

Technical Summary

The basic Leapfrog application software must be developed to communicate with RAM's RAM*first* Gateway and RIM's PageMail Gateway. This will be done by developing two different applications and reusing as much code as possible in both applications.

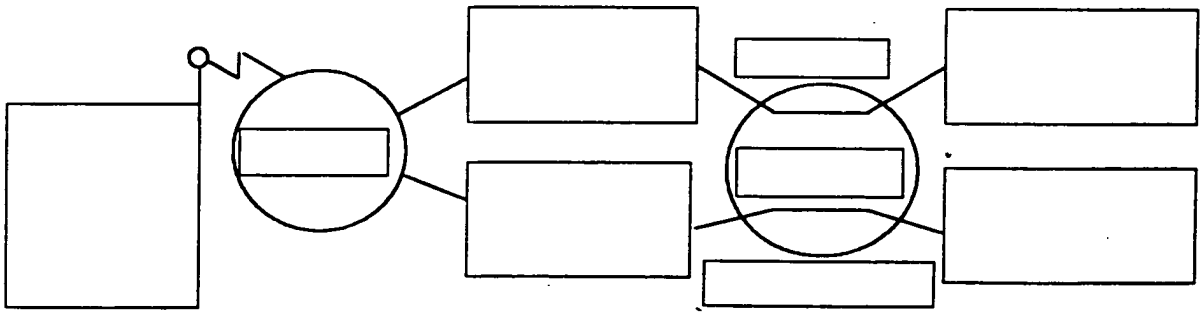
All Leapfrog devices provided to RAM will communicate via the RAMParts Transport Protocol (RTP) to the RAM*first* Gateway. The RTP transport protocol is very similar to the MDP transport and therefore they will replace either other nicely. It will provide delivery of pages and messages beyond 512 bytes, and RIM will be adding a configuration command that will allow the user to set the maximum packet size that can be delivered to the Pager.

The following simple diagram depicts the end-to-end solution:



Connecting to the RAM*first* service will provide the customer with a range of services, depending on which services are connected to any given RAM*first* Gateway. This document will not discuss the services specifically, but will describe how the user accesses those services to achieve the benefits they offer.

The transport method used within the PageMail server is RIM's MDP method. The PageMail server can either be run directly connected to Mobitex via X.25, or via the Packet Blaster using TCP/IP over the Internet. Additionally the PageMail server can also support communication directly between the OutReach product and the Leapfrog.



PageMail will not initially support FAX or other Exchange Server transport methods directly. However in time there will be a series of additional routing option supported.

 ©©) Research In Motion Limited, 1997

General Requirements Overview

Due to delivery schedules, release 1 of Leapfrog is primarily focused on RAM's requirements. As such release 1 only contains a portion of RIM's overall requirements for the Leapfrog. The following is a quickly summary of message delivery and back end support:

RAM Application

- RAM's application will have a minimal RTP protocol implementation with no header support, and if possible support for one configuration packet
- The main purpose of the configuration packet sent by RAM's application is to change the maximum packet that can be exchanged between the Pager and the final destination
- The user will have the ability to configure a range of notification behaviors, depending whether the device is in or out of the holster.

RIM Application

- RIM's application will use a MDP protocol with a compressed MIME (CMIME) data format and support for one configuration packet
- The purposes of the configuration packet sent by RIM's application is to change the maximum packet size and provide a signal to OutReach to start mail redirection
- RIM's application will also have proprietary compression and encryption algorithms and will use a private key from end-to-end

The following are also considerations for back end support:

- Version 1 of RIM's application will work with PageMail as it stands today with minor changes for CMIME support
- Version 1 of RIM's application will work with OutReach with improvements and enhancements to OutReach, these include:
  - Richer integration with Outlook and MAPI
  - Ability to redirect mail, calendar and contacts
  - Support configuration packet for setting maximum packet size and redirection turn on

The following are major functional components that have been discussed but will not be present in Version 1.0 of the Leapfrog or PageMail/OutReach solution:

- Complex table support in the User Interface (UI) Engine
- Multiple game support and a complex calendar application will not be present; these also require complex tables, which are not available until Version 2
- Having added CMIME formatting in Version 1 will allow the addition of forms and scripting support
-

 ©©) Research In Motion Limited, 1997

Leapfrog Key Assignments

The following section illustrates both Leapfrog keyboard assignment and special 'unmarked' Hot Key assignments. The Hot Key assignments are more flexible than the keyboard assignments because they are not tied to markings on the keyboard.

Hard Key Assignment

The following is an illustration of the Leapfrog keyboard assignment and symbol key assignment. The main goal was to keep the calculator keys close to the ALT key and to select the most common symbol keys to be on the keyboard.

Soft Key Assignment

The following is the initial hot key assignment for performing special actions for advanced users.

Keys	List Field	Text Field	Edit Field	Choice Box	Multi-Cho	Table
Roll	1 Entry Up/Down	1 Line Up/Dwn	1 Char Up/Dwn		Next/Prev	Up/Dwn 1 Cell
Alt Roll	1 Page Up/Down	1 Page Up/Dwn	1 Char Left/Rght		Next/Prev	Left/Rght 1 Cell
Shift Roll		Word Left/Rght	Word Left/Right			
Back	Delete Entry	Back To List	BackSpace			
Alt Back			Delete Word			
Alt Click			End Current Line			
Alt Space	***Symbols***	***Symbols***	***Symbols***	***Symbols***	***Symbols***	***Symbols***
Alt Enter	***APP***	***APP***	***APP***	***APP***	***APP***	***APP***
Shift Space			Phrase Complete			
Enter	Default Action	Dft Action	Default Action	Dft Action	Dft Action	Dft Action
Top	Alt G	Alt G	Alt G	Alt G	Alt G	Alt G
Bottom	Alt V	Alt V	Alt V	Alt V	Alt V	Alt V
Shift Click	**Switch To**	**Switch To**	**Switch To**	**Switch To**	**Switch To**	**Switch To**

Bold Italic : O/S or System Key and not handled by UI Engine — Normal: UI Engine Key all Apps.
Underline : Application Specific – optional behaviour — **APP** : Application JUMP key

Stage 1 Requirements

The following is a section by section summary of the Leapfrog functionality being planned for Stage 1 Release 1 for RAM and Stage 1 Release 2 for RIM. The first two sections are common to all other sections, these are the User Interface and Options or System sub-system.

User Interface (UI) Sub-System

The following table is a summary of all UI calls within the UI class library. All calls are C++ member functions within the UI sub-system managing all screen and keyboard control. However, it is important to note that even though the UI Sub-system acts on most keystrokes, it does give the currently executing application first chance action on the key. All RAM functionality implemented in version 1, will also be in version 2 of RIM's application.

RAM User Interface	RAM Functional Description
Lists	A List Box is a read-only list of choices that one item can be selected from and then dismissed
Menus	A Menu List is a read-only list of choices that many items

 ©©) Research In Motion Limited, 1997

	can be selected from, and it must be manually dismissed
8 Line Mode	In the first Release 8 lines of text will always be displayed on Leapfrog's screen; as opposed to 4 or 6 lines
Dialog Box	A dialogue box is displayed as a notice or warning and must be dismissed by the user
Text Box	A read-only field with text data
Edit Box	A read-write field that allows full editing capabilities for information input
Choice Box	A dialog box that allows the user to make a selection between several choices
Multi-Choice Box	A dialog box that allows one line to contain multiple choice boxes, i.e. date and time field
Status Box	A stationary informational box, like a title that does not move
AutoCaps Vs KeyRepeat	User can select between auto capitalization and key repeat; auto cap will take place by holding the down
Preferences Screen	The UI sub-system will expose a series of options through the Options and System menus.
RIM User Interface	**RIM Functional Description**
Simple Tables	A simple tabular data set that allows the user to maneuver through a set of fields
Cut/Copy/Paste	The User will have the ability to enter a Mark Text mode and cut or copy text into an edit buffer – across all applications **Note**: Cutting and Pasting will only be allowed in an editable field since these are destructive actions

 ©©) Research In Motion Limited, 1997

Options Sub-System

The following table is a summary of all Options and System functionality planned for Stage 1. It is important to keep in mind that certain areas like Other Application Configuration and Backup/Restore involve implementation routines from other sub-systems. This is because the options is simply a single interface method for these two important functions. All functionality described in Version 1 of RAM's application will also be in Version 2 of RIM's application.

RAM Options	RAM Functional Description
Options API	The options sub-system will expose an API for other application to post special callable routines
Other Application Configuration	Each application that has options to be changed can choose to provide a route to the options sub-system for centralization
System Wide Backup/Restore	As with the Inter@ctive Pager only the phrase book and address book can be backed up or restored through serial port
Date & Time	There will be options for setting the time and data on the pager
System Status	The system status menu shows the coverage, ESN, MAN number, Battery level, O/S and Application version numbers
Device	View & Modify device settings, including basic notification, contrast, key tone, transmitter on/off, and screen saver timeout period
Auto On/Off	Like the Inter@ctive Pager there will be Automatic On/Off options, including weekend operation
Switch To	The system menu will manage the 'Switch To' menu options for jumping between sub-systems
Symbol Table	The options sub-system will manage the symbol table and the pressing of the symbol key across all sub-systems
AutoCaps ON/OFF	The Auto Caps behaviour will be implemented by the options sub-system and allows certain keystrokes to cause automatic capitals.
TURN OFF	The options sub-system will manage the Turn Off keystroke as entered by the user
BACK LIGHT	The options sub-system will catch the back light keystroke and turn the back light on and off
Application Options	A single entry point into all applications loaded onto the device for configuring different sub-system's options
Network Configuration	As with the Interactive Pager the over-the-air configuration software should be able to work without changes
RIM Options	**RIM Functional Description**
System Wide Backup/Restore	Each application that has a database that can be backed-up or restored can provide a routine to the options sub-system
MESSAGE SIZE & Redirection Turnon	This option allows the user send a configuration message to set their maximum receive size and to start mail redirection
System Wide Backup/Restore	Same backup/restore over the serial port with support for 'all' applications, including e-mail and eventually calendar
Advanced Pager Status	The Advanced Pager status screen will display the state of each of the sub-system, similar to the Palm Pilot, shown the memory used by each and the memory available (assisting with memory decisions)
Holster/Non-Holster Notification Support	There will be advanced configuration settings for guiding how the pager notifies the user when inside and outside the holster

 ©©} Research In Motion Limited, 1997

Message Security	The ability to Enable PIN Protection on the device

From the user's point of view and RAM's point of view, the Leapfrog will act similar to the Inter@ctive Pager, except that they can select the notification mode differently when inside and outside the holster. These settings include:

- **Vibrator Only Mode**: This mode causes the device to vibrate when a new message is received.
- **Beep Only Mode**: This mode causes the device to beep when a new message is received.
- **Beep and Vibrate Mode**: This mode causes the device to beep and vibrate when a new message is received.
- **No Notification**: The user can run the device in silent mode so that it neither beeps nor vibrates.

In addition to being able to set holster and non-holster mode differently, their will be two levels of notification for when a new message arrives in and when messages remain on the device unread. This alert mode is a 'wake-up' to the user in case the first notification was missed.

located in the Message View menu, not in the Options Sub-system. Global settings like alarm, contrast and others will be placed in a "general" Options Sub-system section.

The other main goal of the Options Sub-System is to combine the original system and application options into one options area. The Inter@ctive Pager has two separate sets of options, accessed through the application and the operating system; in the Leapfrog, these options will be combined. The sheer number of options will be reduced by moving options to their respective sub-systems and by carefully managing which options the user sees. For example, the Leapfrog may have an 'Advanced Options' switch that will be turned off by default.

The options the user will select from will include:

- **Time**: Allows the setting of system time, the setting of timers to turn the Pager off and on daily and the setting of a one-time alarm.

- **Device**: Allows the setting of contrast, key clicks, notify modes, security level and turning the radio on and off.

- **Information**: Shows the user general information about the device, like the current radio coverage, device number (MAN), Password security level, serial number, Mobitex Network and Base ID, O/S Revision and Software Revision numbers.

Application Options: Within the application options area will be all application sub-systems that can be configured. This will allow the user to set confirmation on message delete, delete period on old messages and Gateway addressing options. There will also be options for the address book, phrase book and calendar when it is complete.

 ©©) Research In Motion Limited, 1997

E-Mail Sub-System

For every message that arrives in the user will receive the configured notification. The user will be able to immediately press a key and see the message causing the notification alarm. To distinguish between read and unread messages a special indicator will be shown on the list screen. Messages can range in size from 1 character to an undefined upper limit, somewhere in the 10K to 15K range for incoming messages. This can be configured by the user in either the options or Email sub-systems. The Outbox will be able to show a simple indicator as to what has happened to transmitted messages, i.e. pending, sending and sent. The following is a brief list of functions within the E-mail sub-system:

RAM E-Mail	RAM Functional Description
Message Interaction	Messages will be received with a minimal implementation of RTP, message lists will have a title by date with folder being viewed; message headers will not be supported
Messaging API	The message sub-system will export an API so that other sub-systems can send messages, i.e. address book entries
Attachments	The message sub-system will be able to handle a small set of attachments, i.e. address and phrase book entries, exactly the same as used in the Inter@ctive Pager.
Folders	There will be three static folders for mail, the Inbox, Outbox and Saved folders
Insert Key functionality	When reading messages the user can press a key (ALT I) to insert the e-mail address into the address book
Holster Functionality	When in idle mode, and the user pulls the pager out of the holster, they will be taken directly into the message for reading
Advanced Compose	When composing a message multiple TO:, CC: and BCC: fields can be entered for addressing purposes
Customize	The user will be able to customize how information in a folder is displayed; subject and time can both be removed or added to the list view **(See note 1 below)**
Message Management	Hundreds of messages will be stored but to avoid running into capacity limits message will be automatically deleted based on a configured time period; the saved message folder will not be deleted however as are the Inbox and Outbox
Backup/Restore	The ability to backup and restore the E-Mail database must exist – this will be handled by providing an interface to the Options Sub-system
RIM E-Mail	**RIM Functional Description**
Message Interaction	Messages will be received with the latest MDP protocol from RIM; the message list screen will look similar to RAM's view with date, time, title and the folder being viewed
Attachments	Attachments received by the messaging sub-system will be based on CMIME; these will be handled uniquely and be able to immediately spawn the target application when no viewable ASCII text is present in the message
Compression	When working with RIM's Host products all messages will be send and received in compressed format
Encryption	When working with RIM's Host products all messages will be send and received using encryption, with a key of 0
Folders	Since the calendar application will not be immediately available

 ©©) Research In Motion Limited, 1997

	at this stage the addition of a 'Notes' folder would be useful
Backup/Restore	RIM will support the advanced ability to backup and restore the messages on the device
Reply To: field support	When working with RIM's Host products a 'reply to' field will be accepted, and used for addressing replies
Filters	A simple filter can be set up within a folder, i.e. show me all messages from a given e-mail address in this folder
Advanced Gateway	Ability to set a primary and secondary Gateway address; if the secondary gateway address is non-zero then even numbered MAN devices will send to the primary and odd numbered MAN devices will send to the secondary. If a returned packet is received from the current number the send logic will temporarily switch to the alternative number.

The Leapfrog will be capable of holding hundreds of typical length messages. The user needs to avoid exceeding memory 95% full, since this condition will cause excessive "garbage collection" and Flash erase cycles. To avoid this situation the Leapfrog will have several memory management features:

- Encouraging the user to set a maximum message size of between 4K and 10K bytes.
- Automatically deleting old messages; the age limit will default to 1 week. The user will be able to fine-tune this value but not remove the behavior. The range that is being proposed is 1 to 3 weeks, where a target number of messages stored at any one time is between 25 to 50 messages, depending on size.

The normal user would normally read and delete received messages. This step alone would help avoid the need for purging information. The Leapfrog user will also have the option of moving a received message in the Saved area. However, the message management strategy described above would apply to all message areas except the saved message area. The saved message area either would wait 'twice' as long before purging or would have a maximum saved message limit.

The following is a brief description of all the message areas on the Leapfrog:

- **INBOX**: The message view for incoming messages both read and unread. Messages that are unread have a special indicator that is removed once a message is read. Message can be moved from this view to the 'Saved' view if necessary.
- **OUTBOX**: The message view for outgoing messages. These have indicators to show whether they have been sent successfully, whether they are pending or confirmed.
- **SAVED**: The message view for important information to be saved by the user. These messages could be reminders, messages to be sent at a later time or daily reminder messages.

There will be a method for automatically deleting messages from the **saved** message area, based on a combination of message age and the number of messages in the area. This may or may not be a configuration item. (The user does not want to be overloaded with configuration options.)

Messages can be created on the Leapfrog and sent through the Gateway to a variety of destination services. A message can be created in several ways:

- **Self-Authored**: The user creates a message from scratch and either uses an address and addressing method from the address book, or creates an address dynamically for a one-time transfer. The user can address the message to multiple destination addresses or include addresses on a cc: list.
- **Reply**: The user can reply to a received message, which will automatically use the sender's address for the destination. The default address can be changed or added to if the user desires, or a cc: list can also be added to the message reply. The original message contents will be sent with the

 ©©) Research In Motion Limited, 1997

message. If a message comes in from another pager it should go back out the same path, i.e. directly to the pager.

- **Forward**: The user can forward a message which is similar to reply, except the original address is not kept.

After the message is constructed for transmission, the user normally will immediately issue a 'send' command to cause transmission to start, but other options exist for the user. The user can also save the message for later transmission. The user also has the ability to resend the message later, in case of a transmission failure or packet return.

During the transmission of a message, most users are concerned with the progress of the transmission. This concern will be satisfied by adding an indicator to the message, similar to the Inter@ctive Pager. The user will be able to check the Outbox and see changes to the message as it reaches the network. The Outbox message list will show several status indicators, these include:

- **Message Pending**: The message is still pending to be transmitted, either because other messages are queued before it or because there is currently no coverage.
- **Message Sending**: The message is currently being transmitted. It has been given to the radio component and the application is waiting for the 'sent-to-network' indicator.
- **Message Sent**: The message has been sent to the network and for 99.9% of messages, it will make it to the Gateway. If the Gateway is down or congested the packet will get returned and this will be shown.
- **Message Returned**: A transmitted message has been returned by the network. This is because either the FST was down, a network congestion error occurred or the mobile is not activated.
- **Message Mailboxed**: Normally used when sending peer-to-peer messages. This indicates that the mobile being sent to is currently out of coverage and the page has been mailboxed.
- **Message Confirmed**: If supported by the gateway, a confirmation message has been received. This could indicate the message has been submitted to the destination media (Fax, One-Way Pager or SMTP Internet Mail), or that the Internet mail message has reached its destination.

If the Gateway does not support the delivery confirmation then the final message display type will not be supported.

Note 1: There will be wide range of options configured options in this section. Most or all of the Inter@ctive Pager's options will be present including: Confirm Delete, Include Text on Reply, View To/From field, View Time field, Gateway Address, Confirm Delivery and Mailbox Flag.

Note 2: It should be noted that the recent change by RAM to support PIN numbers for peer-to-peer transmission must also be implemented in Leapfrog. It should also be noted that peer-to-peer messages may also be followed by a range of status message from the Gateway as to the progress of message delivery. These indications should be handled as they are today in the Inter@ctive Pager.

Note 3: There are several questions that still are outstanding as we move to the RTP protocol. One question to be resolved is whether Extended HP-ID 4 Specification apply when using the RTP protocol in the RAMfirst gateway?

 ©©) Research In Motion Limited, 1997

Address Book Sub-System

The address book will be similar to the Inter@ctive Pager today. The user will be able to added, change and delete address book entries, which are used primarily for authoring and sending mail to other people.

RAM Address Book	RAM Functional Description
Address Book API	The addresses book with export functions for all other sub-systems to call, mainly for compose and attachment
Recent List	Most initial address book lists show a cache or most frequently accesses addresses, cache size is configurable
Full List	The normal view will be a complete 'sorted' list of all addresses, also called the full list
Search On View	Within each view of the address book (recent or full list), the user can search or narrow the list by setting up a search criteria to create a subset of the current list
Select Process	The selection process involves picking a name, and then picking a 'method' for message delivery. When there is only one method, the later step will be skipped
Add Addresses	The user will be able to addresses with a fixed number of elements per address
Add Addresses Programmatically	Other programs will be able to addresses; perhaps via attachments
Edit Address Data	The user will be able to view and edit address data using the address book sub-system
Sort Views by first name	The user will be able to sort the Recent and Full view by a one of three methods, including : first name, last name and company name
Backup/Restore Options	Connection to the Options sub-system to do backup and restore of address book database
RIM Address Book	**RIM Functional Description**
Add Addresses and User Defined Fields	The user will be able to addresses and personal fields to the address entry; limits on new fields will exist
Add Addresses Programmatically, including Fields (keys, etc)	Other programs will be able to addresses and special fields programmatically; encryption keys may be necessary if we go to public keys
Address Book API	The addresses book with export functions for all other sub-systems to call, mainly for compose and attachment
Collapse Large Lists	The ability to collapse large address lists and use a 'title' only method for reducing long menu lists
Customize	The user can customize the address book, primarily in the number of cached items in the recent list, the new fields added to each address book entry and number of items before collapsing take place

 ©©) Research In Motion Limited, 1997

The address book is a database of names that can be used for general reference or for construction of mail messages. The current list of the default fields supported in the address book:

- **Name**: Both the first and last name of the person being referenced.
- **E-Mail**: E-Mail address for sending a message via SMTP to the Internet.
- **Fax**: Fax number for sending a message to a destination Fax modem.
- **Phone**: A phone number for sending a message, via text-to-voice conversation to a destination phone number. This is normally just for reference and holding contact information.
- **One-Way**: A one-way pager PIN number for sending a message to a one-way pager
- **Two-Way**: A Leapfrog Pager MAN/PIN or Inter@ctive Pager MAN/PIN that will be used for sending a message through the gateway to another Pager. The Gateway will be responsible for converting from RTP to HP-ID 4 for the delivery and for PIN resolution to a MAN.
- **Address**: the actual physical address of the person referenced in this address book entry.
- **Misc.**: Some number of remaining elements, perhaps a Note, Cell Phone number and Home Phone Number. The final selection of these has not been determined at this time.

The overall functionality within the address book sub-system will be similar to that of the Inter@ctive Pager. The user will be able to create, modify and delete address book entries up to 750 entries in total. When the user creates a message they will be taken directly to the address book to select a destination for the message. The user must enter at least one of the routing fields when creating an address book entry.

Note 1: The ability to dynamically add fields to the address book is a RIM feature that should be only considered in the release 2 of the Leapfrog product.

 ©©) Research In Motion Limited, 1997

Phrase Book Sub-System

The phrase book is similar to the message book within the Inter@ctive Pager. It contain frequently used phrases that can be inserted into mail messages to save time in typing messages.

RAM Phrase Book	RAM Functional Description
Phrase Book API	The phrase book exports an set of API calls so that the other sub-systems can make use of the phrase book
Categories	The phrase book will place phrases into categories, the use has the ability to add categories
Transmit/Display Strings	Each phrase in the phrase book can be set up to display one string and transmit a different string
Editable/non-editable categories	Some categories can be setup to be non-editable; this protects some phrases from being changed or deleted
Phrase Completion	When in an edit field, like the message compose screen, the user can press a key to have a phrase completed automatically
Phrase Sorting	The ability to type part of a phrase to narrow the presentation list for selection of a phrase
Backup/Restore Option	The ability to post a backup/restore routine to the Options sub-system to save and restore the phrase book database
RIM Phrase Book	**RIM Functional Description**
Phrase Book API	The phrase book exports an set of API calls so that the other sub-systems can make use of the phrase book
Customize	The user can customize the phrase book by setting up the following parameters
Phrase Completion (multi-word)	The phrase completion method can also substitute multiple words for a single 'short cut', i.e. RIM converts to Research In Motion
Programmatic Adding of Phrase Categories and String	The ability for another program to add phrase programmatically like the address book or calendar

The Phrase Book is similar in concept to the Address Book, but contains a database of strings. These strings are used to speed up message entry by allowing the user to insert the strings into actual messages to be transmitted. These short-cut strings are sorted into categories, similar to those used on the Inter@ctive Pager. Some of these categories include:

 ©©) Research In Motion Limited, 1997

- **Appreciation**: Appreciation, gratitude and thank-you based phrases messages and statements, which are normally used to open or close messages.

- **Closings**: A collection of strings that are considered closings or terminating messages in a mail message.

- **Greetings**: A collection of strings that are considered greetings or opening statements in a mail message.

- **Miscellaneous**: Any other strings that are added to the system for insertion into mail messages.

There will be other categories; a full list will be decided closer to completion of the Phrase Book sub-system. The user will be able to add, delete and edit categories as they can on the Inter@ctive Pager today.

The user will be able to create, modify and delete phrase book strings, up to 750 phrases in total. When creating a phrase book entry, the user will be able to type messages directly into the Phrase Book under a given category. Every string must be placed into a category by definition, there will be no way to enter a string without entering a category.

 ©©) Research In Motion Limited, 1997

Additional Sub-Systems

These additional sub-systems are not expected by RAM, but RIM would like to include them into their service offering. This means that they may not be included in RAM software release, but they will be a part of the RIM software service offering.

RIM Other Applications	RIM Functional Description
Basic Calendar	Provides a month at a glance view with no provisions for notes, or events (scheduler functionality is not supported)
Alarm Clock	A basic alarm clock that can be set to go off at any time on any date. Uses one of the preset **Tunes**.
Calculator	A simple calculator to turn the pager into a simple PIM functionality; compete better with competitors

The Stack

The RIM stack must support both RIM and RAM's protocol requirements.

Transport	
RTP	RAM's protocol stack – replacing the original HPID 4
Transport MDP	RIM's internally defined protocol stack
Compression	
Encryption	Encryption with a key of 0 initially (RIM only)
Datagram API	The stack exports an set of API calls so that the applications above can access the network. **For future expansion.**
Message Formatting (CMIME)	Special MIME format from Outreach or from PageMail that is formatted in compressed format for Pager to interpret

 ©©) Research In Motion Limited, 1997

Desktop Support Software

To assist in the support and maintenance of the Leapfrog there will be a desktop tool for the configuration and management of the Leapfrog. This product should be similar or exactly what RAM has today, which means the Leapfrog software should be compatible with the software protocols.

Basic Desktop Support

To satisfy basic requirements for RAM and for low-end RIM Pager users a basic configuration program is required. Some of the functions in this piece of this software will go beyond the current software and will include:

- **Advanced Configuration**: Secure configuration for network operator level features; there may be very few of these for Leapfrog since it cannot be used as a normal modem.
- **Network Configuration**: Allows a service provider to set options like gateway address and encryption keys within the pager.
- **Basic User Configuration**: User based configuration control to allow the user the ability to set certain parameters and make working with the device easier.
- **Off-line Backup**: This new feature will allow the user to store all their address and phrase book information in an off-line location. This allows for maintaining a backup of information on the Leapfrog, or quickly setting up a new Leapfrog with custom address and phrase books.

The current plan is to restrict what parameters can be updated over the Mobitex network. Some parameters, password, backup and restore of address and phrase book will only work when the device is connected locally via the serial port of the desktop computer. Most Leapfrog users will not have a secondary radio modem to perform the updates over the air, and would want to update their message store locally via a serial modem.

Specific application parameters, like the Gateway address, can be updated over the Mobitex network. To perform this update, a security password will be set up for each of the three configuration levels. This password will be maintained between the Leapfrog and the desktop configuration package that will be managed by different parties, i.e. user, service provider and network operator. This is similar to the remote configuration packet being used with the Inter@ctive Pager today.

Advanced Desktop Support

For those uses that are using a product that supports Microsoft's Messaging API (MAPI), they have the option of using Outreach at their desktop. Outreach is a mail redirector and backup/restore utility combined. It provides to the user a bi-directional path between their pager and their desktop. This allows mail, contact and schedule information to be sent to the user and for mail to be sent back from the user. In later stages contact and schedule information will also be sent back to the desktop with mail. The basic functional of Outreach includes:

- Setup and configuration of pager and environment for redirection, including initial download of selected personal and global addresses from MAPI message store.
- Ability to start redirection either through an idle period, like a screen save kickoff, or through a signal from the pager to the desktop. Redirection can be filtered by the user based on the user-id and other criteria like size and preferred/non-preferred addresses.
- A log file is kept of all message sent and received through Outreach, which can be reviewed later when the user comes back to the office.
- A desktop icon can be used by other employees in the office to contact the user while on the road. They can send messages and receive responses directly to the pager icon.
- A set of information can be redirected, including messages, address information and schedule information. With the release 1 application only message and addresses can be imported to their

 ©©) Research In Motion Limited, 1997

appropriate sub-systems, the calendar information will simply stay as a mail message.

Stage 2 Requirements

All stage 2 requirements are targeted at the RIM Leapfrog application. It is not expected that the RAM application will change much and RIM will not be developing 'spontaneously' for RAM.

UI	Description
Complex Tables	Tables with advanced formatting options – supporting different fonts, bitmaps (?). Required for TK and some of the games
6 Line Mode	A display mode that supports a larger font.
ALT Indicator	An indicator displaying the state of the ALT key – this is more of a system design issue that should possibly be solved now.
Shift Indication	An indicator displaying the state of the Shift key – this is more of a system design issue that should possibly be solved now.
CAPS Indicator	An indicator displaying the state of CAPS lock – this is more of a system design issue that should possibly be solved now.
Password Edit Box (******)	A dialog box that handles password hiding (the *'s instead of the actual characters
Alignment of Multiple Edit Fields	Visually line up fields and labels.
Justification of Edit	Left, Right, and Centered justification
Radio Buttons	A method of grouping multiple buttons, where only one can be selected.
Selection	Able to mark a section of text for modification of some sort. Initially only deletion or replace (marking a section and typing a character replaces that selection.
Check Box	A box that can be marked as selected

Options	Description
Security	Addition of different security levels, initially in Phase 1, Release 2 security is simply turned on and off!
Key download & protocol	Although the device had encryption, it used a key of zero; enables downloading of keys for both the service operator and the desktop user

Email	Description
Forms	Support for a portion of HTML 4.0 – unsupported formatting is either ignored or filtered by the host.
Scripting	A subset of the Visual Basic command set.
Free Form Addressing	Typing of a free form address on the To: line – requires prototyping before look and feel issues can be solved.

 ©©) Research In Motion Limited, 1997

Address Book	Description
Sort on other....	Sorting on fields other then first name, last name or company
Import Addresses – view attachments	Support for address entries sent by attachment
Import addresses – restore	Restore addresses via the Backup/Restore utility – **MOVE TO V1**
Export Addresses – transmit attachments	Support to sending addresses via attachments
Export Addresses – backup	Backup addresses via the Backup/Restore utility – **MOVE TO V1**
Address groups	Local grouping of addresses on the pager.
Search for remote address info	
One Time addresses	Ability to use an address without adding it to the address book on a permanent basis
Copy/Paste of address objects	Copy and paste addresses as an object (into groups, messages, etc)
Collapse/Expand of address lists	Collapse sections (the 'D's) of the list when there are more then a set number of them. May not be required since same essential functionality is provided by using the search capabilities of the database to limit the view.

Phrase Book	Description
Nested Categories	Categories can contain sub-categories
Invisible categories	Categories not visible to the user
Invisible phrases	Phrases not visible to the user
Non-editable phrases	Read-only phrases
Auto-Correct	A limited dictionary that can be used to convert typing mistakes such as *taht* to *that*. Can also be used as a shortcut to phrase expansion: *wrt* expands to *with respect to*
Intelligence (auto-add phrases)	Applications can build up the phrasebook dictionary automatically depending on the text the user enters in sent messages (or from other sources)

TK	Description
Calendar	A full featured calendar supporting event indication: different fonts for dates with events, tasks, notes, etc. Supports direct links to Task List, Scheduler, Notepad, and Journal below.
Task List	As above
Scheduler	As above
Note Pad	As above
Journal	As above
Outreach	Full support for Outreach remote scheduling

 ©©} Research In Motion Limited, 1997

Games And Other Apps	Description
Chess	Two player
Battleship	Two player
Connect 4	Two player
Calculator	Two player

Stage 3 Requirements (or later)

Public Key Encryption is the currently the only foreseen development in stage three. Adding support for public key encryption requires changes to the stack, the messaging sub-system, and possibly the address book as well.

Further stage 3 requirements will be added as the product matures, and feedback about version 1 (and possibly version 2 as well) is returned.

 ©©) Research In Motion Limited, 1997

$/Pagemail/Redirector/Server/main.cpp

****
Label: Build 1.6 RC1b
User: Alewis Date: 5/18/99 Time: 1:01p
Labeled 'Build 1.6 RC1b'
Label comment:
adds auto-signature

****
Label: Build 1.6 RC1a
User: Alewis Date: 5/14/99 Time: 5:40p
Labeled 'Build 1.6 RC1a'
Label comment:

********** Version 68 **********
User: Alewis Date: 5/06/99 Time: 4:53p
Checked in $/Pagemail/Redirector/Server
Comment:
go back to old initialize and shutdown methods - it broke in 95/98.
We will have to use a different method of ensuring the same thread is used in NT service

****
Label: Build 1.6 RC1
User: Alewis Date: 5/04/99 Time: 7:58p
Labeled 'Build 1.6 RC1'
Label comment:

********** Version 67 **********
User: Alewis Date: 5/04/99 Time: 5:48p
Checked in $/Pagemail/Redirector/Server
Comment:
move SCS initialization and uninitialization to within the threadproc so that we can guarantee that they are called on the same thread.

****
Label: Build 1.6.0 Desktop
User: Alewis Date: 4/30/99 Time: 12:31a
Labeled 'Build 1.6.0 Desktop'
Label comment:

********** Version 66 **********
User: Alewis Date: 4/29/99 Time: 9:11p
Checked in $/Pagemail/Redirector/Server
Comment:
updated version numbers

****
Label: Checkpoint pre-RC1 Server
User: Alewis Date: 4/29/99 Time: 1:03a
Labeled 'Checkpoint pre-RC1 Server'
Label comment:

****
Label: Checkpoint April 5
User: Alewis Date: 4/05/99 Time: 6:19p
Labeled 'Checkpoint April 5'
Label comment:

```
**********
Label: Server Build Beta 2b
User: Alewis          Date:  4/01/99  Time:  2:59a
Labeled 'Server Build Beta 2b'
Label comment:


**********
Label: Server Build Beta 2a
User: Alewis          Date:  3/26/99  Time: 11:04a
Labeled 'Server Build Beta 2a'
Label comment:


**********
Label: Server Build Beta 2
User: Alewis          Date:  3/24/99  Time:  6:16p
Labeled 'Server Build Beta 2'
Label comment:


**********  Version 65  **********
User: Alewis          Date:  3/24/99  Time:  6:16p
Checked in $/Pagemail/Redirector/Server
Comment:
  update -? help

**********
Label: Database Change - Interim Label
User: Cdunk           Date:  3/21/99  Time:  3:09p
Labeled 'Database Change - Interim Label'
Label comment:
  Changes to database are made, integrated into all the projects
in the server DSW, but not the database utilities or the pocketlink projects.
Get this label to build the pocketlink or database utility projects.

**********
Label: Server Build Beta 1a
User: Alewis          Date:  3/05/99  Time:  2:01p
Labeled 'Server Build Beta 1a'
Label comment:


**********
Label: Server Build Beta 1
User: Alewis          Date:  2/26/99  Time:  5:04p
Labeled 'Server Build Beta 1'
Label comment:


**********
Label: HostSDK 1.1
User: Cdunk           Date:  2/23/99  Time:  1:46p
Labeled 'HostSDK 1.1'
Label comment:
  Same as HostSDK 1.0 with a slightly different project
structure.

**********
Label: Host SDK Install 1.0
User: Jsauer          Date:  2/15/99  Time: 11:44a
```

Labeled 'Host SDK Install 1.0'
Label comment:

****
Label: Server Snapshot for Descartes
User: Alewis Date: 2/11/99 Time: 2:00p
Labeled 'Server Snapshot for Descartes'
Label comment:

****
Label: Build 1.5.0c
User: Alewis Date: 2/08/99 Time: 6:05p
Labeled 'Build 1.5.0c'
Label comment:

****
Label: 1.5.0b
User: Alewis Date: 1/29/99 Time: 5:03p
Labeled '1.5.0b'
Label comment:

****
Label: Build 1.5.0a
User: Alewis Date: 1/12/99 Time: 4:52p
Labeled 'Build 1.5.0a'
Label comment:

********** Version 64 **********
User: Alewis Date: 1/11/99 Time: 10:41a
Checked in $/Pagemail/Redirector/Server
Comment:
change Health checks to return boolean; periodically automatically check the "health" of the worker threads

****
Label: Build 1.5.0
User: Alewis Date: 1/04/99 Time: 2:32p
Labeled 'Build 1.5.0'
Label comment:

****
Label: Build 1.4.10
User: Alewis Date: 12/21/98 Time: 11:47a
Labeled 'Build 1.4.10'
Label comment:

****
Label: Build 1.4.9
User: Alewis Date: 12/16/98 Time: 2:37p
Labeled 'Build 1.4.9'
Label comment:

****
Label: Build 1.0.8.1
User: Alewis Date: 12/11/98 Time: 5:02p

Labeled 'Build 1.0.8.1'
Label comment:

****
Label: Build 1.0.8
User: Alewis Date: 12/10/98 Time: 6:19p
Labeled 'Build 1.0.8'
Label comment:

****
Label: Build 1.0.7
User: Alewis Date: 12/07/98 Time: 5:45p
Labeled 'Build 1.0.7'
Label comment:

****
Label: 1.0.7 Rogue 1
User: Alewis Date: 12/04/98 Time: 4:56p
Labeled '1.0.7 Rogue 1'
Label comment:

****
Label: pre-imap relay
User: Cdunk Date: 12/02/98 Time: 2:07p
Labeled 'pre-imap relay'
Label comment:
relay and the code it depends on prior to imap integration and tertiary changes

********** Version 63 **********
User: Alewis Date: 12/02/98 Time: 12:46p
Checked in $/Pagemail/Redirector/Server
Comment:
update copyright notices

********** Version 62 **********
User: Alewis Date: 11/30/98 Time: 11:55a
Checked in $/Pagemail/Redirector/Server
Comment:
change PocketLink to BlackBerry; new registry usage

****
Label: Build 1.0.6
User: Alewis Date: 11/17/98 Time: 3:44p
Labeled 'Build 1.0.6'
Label comment:

****
Label: Build 1.0.5a
User: Alewis Date: 11/16/98 Time: 5:23p
Labeled 'Build 1.0.5a'
Label comment:

****
Label: Build 1.0.5
User: Alewis Date: 11/12/98 Time: 2:40p
Labeled 'Build 1.0.5'

```
Label comment:


**********
Label: Build 1.0.4
User: Alewis          Date: 11/11/98  Time:  1:55p
Labeled 'Build 1.0.4'
Label comment:


**********  Version 61  **********
User: Srahn           Date: 11/10/98  Time:  4:34p
Checked in $/Pagemail/Redirector/Server

**********  Version 60  **********
User: Srahn           Date: 11/10/98  Time:  9:37a
Checked in $/Pagemail/Redirector/Server
Comment:
  Thread health feature.  Improved shutdown performance.

**********
Label: Build 1.0.3
User: Alewis          Date: 11/09/98  Time:  6:17p
Labeled 'Build 1.0.3'
Label comment:


**********  Version 59  **********
User: Alewis          Date: 11/09/98  Time: 10:26a
Checked in $/Pagemail/Redirector/Server
Comment:
  clean up warnings

**********
Label: Build 1.0.1
User: Alewis          Date: 10/19/98  Time:  5:28p
Labeled 'Build 1.0.1'
Label comment:


**********
Label: Build 1.0.0
User: Alewis          Date: 10/16/98  Time:  6:18p
Labeled 'Build 1.0.0'
Label comment:


**********  Version 58  **********
User: Alewis          Date: 10/16/98  Time: 12:59p
Checked in $/Pagemail/Redirector/Server
Comment:
  update version numbers

**********
Label: Build 0.13.0
User: Alewis          Date: 10/09/98  Time:  1:59p
Labeled 'Build 0.13.0'
Label comment:


**********
Label: Build 0.12.0
User: Alewis          Date: 10/07/98  Time:  3:45p
```

```
Labeled 'Build 0.12.0'
Label comment:


**********
Label: Build 0.11.7
User: Alewis          Date:  9/30/98  Time:  4:45p
Labeled 'Build 0.11.7'
Label comment:


**********
Label: Build 0.11.6
User: Alewis          Date:  9/29/98  Time:  4:39p
Labeled 'Build 0.11.6'
Label comment:


**********
Label: Build 0.11.5
User: Alewis          Date:  9/29/98  Time:  4:02p
Labeled 'Build 0.11.5'
Label comment:


**********
Label: 0.11.5
User: Alewis          Date:  9/29/98  Time:  4:02p
Labeled '0.11.5'
Label comment:


**********
Label: Build 0.11.4
User: Alewis          Date:  9/28/98  Time:  5:25p
Labeled 'Build 0.11.4'
Label comment:


**********
Label: Build 0.11.0
User: Alewis          Date:  9/24/98  Time:  5:41p
Labeled 'Build 0.11.0'
Label comment:


**********  Version 57  **********
User: Alewis          Date:  9/24/98  Time:  4:26p
Checked in $/Pagemail/Redirector/Server
Comment:
  replace some instances of 'PageMail' with 'PocketLink'

**********
Label: Build 0.10.0
User: Alewis          Date:  9/21/98  Time:  2:51p
Labeled 'Build 0.10.0'
Label comment:
  The "real" one?

**********
Label: Build.008
User: Roliver         Date:  9/04/98  Time:  4:13p
Labeled 'Build.008'
```

Label comment:
  Next release of redirector for internal beta.

****
Label: Build.007
User: Roliver Date: 8/20/98 Time: 4:29p
Labeled 'Build.007'
Label comment:
  True Release of Redirector/Relay/PMDB for Internal Beta.

****
Label: Build.006
User: Roliver Date: 8/19/98 Time: 4:21p
Labeled 'Build.006'
Label comment:
  Release for Redirector Internal RIM Beta Test.

****
Label: Build.005
User: Alewis Date: 8/14/98 Time: 3:38p
Labeled 'Build.005'
Label comment:
  (Build .004 was never officially labelled)

****
Label: Build.003
User: Alewis Date: 7/27/98 Time: 3:27p
Labeled 'Build.003'
Label comment:

****
Label: Build.002
User: Alewis Date: 7/24/98 Time: 5:03p
Labeled 'Build.002'
Label comment:

********** Version 56 **********
User: Alewis Date: 7/20/98 Time: 11:16a
Checked in $/Pagemail/PageMail
Comment:
  suppress a compile warning

****
Label: Build.001
User: Alewis Date: 7/09/98 Time: 5:06p
Labeled 'Build.001'
Label comment:
  First official build. Lock and Load!

********** Version 55 **********
User: Roliver Date: 7/09/98 Time: 12:20p
Checked in $/Pagemail/PageMail
Comment:
  Call InitializeSystem with the proper parameters.

****
Label: 0.0.4
User: Roliver Date: 6/01/98 Time: 11:46a
Labeled '0.0.4'
Label comment:
  This label is pre- Personal Pagemail. GME, CMIME and ETP have

been added to the project, though not fully incorporated into usercontrol as of this label.

****
Label: 0.0.3
User: Pagemailadmin Date: 5/06/98 Time: 4:13p
Labeled '0.0.3'
Label comment:
  This is a pre GME/CMIME, Personal Pagemail, Relay Pagemail, more command label... Just in case.

****
Label: 0.0.2
User: Roliver Date: 4/24/98 Time: 3:28p
Labeled '0.0.2'
Label comment:
  Add read/delete/moved notification support - pending DeviceSends are now cancelled on this notifications.
Handle DeviceSend responses: Expired -> immediate retry, RejectedByNetwork -> slow retry, RejectedByDevice & RejectedByPacketblaster -> cancel;
Includes Allans changes to the Utilities, after the code review.

****
Label: Version 0.0.1
User: Roliver Date: 4/21/98 Time: 1:59p
Labeled 'Version 0.0.1'
Label comment:
  Uses a worker thread pool.
EntryId's are used to track new mail.
(Delete and Read cancelling is not yet implemented)
(Send/Receive failures are not yet handled)

********** Version 54 **********
User: Alewis Date: 3/09/98 Time: 5:34p
Checked in $/Pagemail/PageMail

********** Version 53 **********
User: Alewis Date: 3/02/98 Time: 7:55a
Checked in $/Pagemail/PageMail

********** Version 52 **********
User: Alewis Date: 2/25/98 Time: 5:21p
Checked in $/Pagemail/PageMail

********** Version 51 **********
User: Alewis Date: 2/25/98 Time: 4:51p
Checked in $/Pagemail/PageMail

********** Version 50 **********
User: Alewis Date: 2/25/98 Time: 3:58p
Checked in $/Pagemail/PageMail

********** Version 49 **********
User: Alewis Date: 2/25/98 Time: 1:21p
Checked in $/Pagemail/PageMail

********** Version 48 **********
User: Alewis Date: 2/19/98 Time: 4:54p
Checked in $/Pagemail/PageMail

********** Version 47 **********
User: Alewis Date: 2/19/98 Time: 1:07p

Checked in $/Pagemail/PageMail

********** Version 46 **********
User: Alewis Date: 2/19/98 Time: 1:02p
Checked in $/Pagemail/PageMail

********** Version 45 **********
User: Alewis Date: 2/10/98 Time: 3:45p
Checked in $/Pagemail/Server

********** Version 44 **********
User: Srahn Date: 2/05/98 Time: 1:21p
Checked in $/Pagemail/Server

********** Version 43 **********
User: Alewis Date: 1/29/98 Time: 5:37p
Checked in $/Pagemail/Server

********** Version 42 **********
User: Alewis Date: 1/21/98 Time: 2:13p
Checked in $/Pagemail/Server

********** Version 41 **********
User: Alewis Date: 1/13/98 Time: 5:32p
Checked in $/Pagemail/Server

********** Version 40 **********
User: Pagemailadmin Date: 11/17/97 Time: 1:39p
Checked in $/Server

********** Version 39 **********
User: Pagemailadmin Date: 11/07/97 Time: 2:12p
Checked in $/Server

********** Version 38 **********
User: Pagemailadmin Date: 11/06/97 Time: 5:52p
Checked in $/Server

********** Version 37 **********
User: Cdunk Date: 11/06/97 Time: 5:08p
Checked in $/Server

********** Version 36 **********
User: Cdunk Date: 10/31/97 Time: 9:07a
Checked in $/Server

********** Version 35 **********
User: Pagemailadmin Date: 10/30/97 Time: 9:04a
Checked in $/Server

********** Version 34 **********
User: Cdunk Date: 10/29/97 Time: 3:00p
Checked in $/Server

********** Version 33 **********
User: Pagemailadmin Date: 10/28/97 Time: 11:18a
Checked in $/Server

********** Version 32 **********
User: Cdunk Date: 10/24/97 Time: 4:40p
Checked in $/Server

********** Version 31 **********

User: Pagemailadmin Date: 10/24/97 Time: 3:40p
Checked in $/Server

********** Version 30 **********
User: Pagemailadmin Date: 10/24/97 Time: 2:42p
Checked in $/Server

********** Version 29 **********
User: Cdunk Date: 10/24/97 Time: 10:11a
Checked in $/Server

********** Version 28 **********
User: Pagemailadmin Date: 10/16/97 Time: 12:55p
Checked in $/Server

********** Version 27 **********
User: Pagemailadmin Date: 10/16/97 Time: 9:32a
Checked in $/Server

********** Version 26 **********
User: Bgilhuly Date: 10/15/97 Time: 2:44p
Checked in $/Server

********** Version 25 **********
User: Pagemailadmin Date: 10/15/97 Time: 12:56p
Checked in $/Server

********** Version 24 **********
User: Mbrown Date: 10/10/97 Time: 10:14a
Checked in $/Server

********** Version 23 **********
User: Mbrown Date: 10/10/97 Time: 9:21a
Checked in $/Server

********** Version 22 **********
User: Cdunk Date: 10/09/97 Time: 1:18p
Checked in $/Server

********** Version 21 **********
User: Bgilhuly Date: 10/09/97 Time: 9:59a
Checked in $/Server

********** Version 20 **********
User: Bgilhuly Date: 10/09/97 Time: 9:27a
Checked in $/Server

********** Version 19 **********
User: Bgilhuly Date: 10/07/97 Time: 1:04p
Checked in $/Server

********** Version 18 **********
User: Bgilhuly Date: 10/07/97 Time: 12:55p
Checked in $/Server

********** Version 17 **********
User: Bgilhuly Date: 10/07/97 Time: 12:54p
Checked in $/Server

********** Version 16 **********
User: Bgilhuly Date: 10/07/97 Time: 12:53p
Checked in $/Server

```
****************** Version 15 ******************
User: Cdunk          Date: 10/06/97  Time:  1:27p
Checked in $/Server

****************** Version 14 ******************
User: Cdunk          Date: 10/06/97  Time: 12:42p
Checked in $/Server

****************** Version 13 ******************
User: Nomad          Date: 10/06/97  Time: 10:31a
Checked in $/Server

****************** Version 12 ******************
User: Cdunk          Date: 10/06/97  Time: 10:17a
Checked in $/Server

****************** Version 11 ******************
User: Bgilhuly       Date: 10/03/97  Time:  6:44p
Checked in $/Server

****************** Version 10 ******************
User: Bgilhuly       Date: 10/03/97  Time:  3:46p
Checked in $/Server
Comment:
  ready to integrate

****************** Version 9 ******************
User: Bgilhuly       Date: 10/02/97  Time:  6:12p
Checked in $/Server

****************** Version 8 ******************
User: Cdunk          Date:  9/25/97  Time:  2:28p
Checked in $/Server

****************** Version 7 ******************
User: Cdunk          Date:  9/24/97  Time:  6:21p
Checked in $/Server

****************** Version 6 ******************
User: Cdunk          Date:  9/23/97  Time:  5:42p
Checked in $/Server

****************** Version 5 ******************
User: Cdunk          Date:  9/23/97  Time:  4:47p
Checked in $/Server

****************** Version 4 ******************
User: Cdunk          Date:  9/23/97  Time: 10:17a
Checked in $/Server

****************** Version 3 ******************
User: Cdunk          Date:  9/23/97  Time:  9:59a
Checked in $/Server

****************** Version 2 ******************
User: Cdunk          Date:  9/23/97  Time:  9:26a
Checked in $/Server

****************** Version 1 ******************
User: Cdunk          Date:  9/22/97  Time:  3:40p
Created main.cpp
```

$/Pagemail/Redirector/scs.cpp

****
Label: Build 1.6 RC1b
User: Alewis Date: 5/18/99 Time: 1:01p
Labeled 'Build 1.6 RC1b'
Label comment:
adds auto-signature

****
Label: Build 1.6 RC1a
User: Alewis Date: 5/14/99 Time: 5:40p
Labeled 'Build 1.6 RC1a'
Label comment:

********** Version 190 **********
User: Alewis Date: 5/13/99 Time: 5:04p
Checked in $/Pagemail/Redirector
Comment:
desktop redirector wasn't working with new configuration, in which the "server side" was not created in single-user mode

********** Version 189 **********
User: Alewis Date: 5/06/99 Time: 4:53p
Checked in $/Pagemail/Redirector
Comment:
go back to old initialize and shutdown methods - it broke in 95/98. We will have to use a different method of ensuring the same thread is used in NT service

****
Label: Build 1.6 RC1
User: Alewis Date: 5/04/99 Time: 7:58p
Labeled 'Build 1.6 RC1'
Label comment:

********** Version 188 **********
User: Alewis Date: 5/04/99 Time: 5:48p
Checked in $/Pagemail/Redirector
Comment:
move SCS initialization and uninitialization to within the threadproc so that we can guarantee that they are called on the same thread.

****
Label: Build 1.6.0 Desktop
User: Alewis Date: 4/30/99 Time: 12:31a
Labeled 'Build 1.6.0 Desktop'
Label comment:

****
Label: Checkpoint pre-RC1 Server
User: Alewis Date: 4/29/99 Time: 1:03a
Labeled 'Checkpoint pre-RC1 Server'
Label comment:

********** Version 187 **********
User: Alewis Date: 4/29/99 Time: 1:03a
Checked in $/Pagemail/Redirector
Comment:

make it compile for desktop redirector again

********** Version 186 **********
User: Ipatters Date: 4/28/99 Time: 7:06p
Checked in $/Pagemail/Redirector
Comment:

********** Version 185 **********
User: Ipatters Date: 4/28/99 Time: 10:24a
Checked in $/Pagemail/Redirector
Comment:
BESMonitor additions...

********** Version 184 **********
User: Gvuong Date: 4/21/99 Time: 4:56p
Checked in $/Pagemail/Redirector
Comment:
Added parameter in Initialize()

********** Version 183 **********
User: Alewis Date: 4/21/99 Time: 10:13a
Checked in $/Pagemail/Redirector
Comment:
initialization of the Random subsystem

****
Label: Checkpoint April 5
User: Alewis Date: 4/05/99 Time: 6:19p
Labeled 'Checkpoint April 5'
Label comment:

****
Label: Server Build Beta 2b
User: Alewis Date: 4/01/99 Time: 2:59a
Labeled 'Server Build Beta 2b'
Label comment:

********** Version 182 **********
User: Alewis Date: 3/29/99 Time: 4:02p
Checked in $/Pagemail/Redirector
Comment:
extra logging

****
Label: Server Build Beta 2a
User: Alewis Date: 3/26/99 Time: 11:04a
Labeled 'Server Build Beta 2a'
Label comment:

********** Version 181 **********
User: Cdunk Date: 3/26/99 Time: 10:22a
Checked in $/Pagemail/Redirector
Comment:
avoided deadlock by leaving queue unlocked when calling in to the database

****
Label: Server Build Beta 2
User: Alewis Date: 3/24/99 Time: 6:16p

```
Labeled 'Server Build Beta 2'
Label comment:


****************** Version 180 ******************
User: Cdunk          Date:  3/22/99  Time: 10:49a
Checked in $/Pagemail/Redirector
Comment:
  Changes required for new database implementation

****************** Version 179 ******************
User: Cdunk          Date:  3/21/99  Time:  3:26p
Checked in $/Pagemail/Redirector
Comment:
  Changes to make use of new database interface

**********************
Label: Database Change - Interim Label
User: Cdunk          Date:  3/21/99  Time:  3:09p
Labeled 'Database Change - Interim Label'
Label comment:
  Changes to database are made, integrated into all the projects
in the server DSW, but not the database utilities or the pocketlink projects.
Get this label to build the pocketlink or database utility projects.

****************** Version 178 ******************
User: Alewis         Date:  3/08/99  Time:  8:27a
Checked in $/Pagemail/redirector
Comment:
  incorrect iterator used; remove unused variables

**********************
Label: Server Build Beta 1a
User: Alewis         Date:  3/05/99  Time:  2:01p
Labeled 'Server Build Beta 1a'
Label comment:


**********************
Label: Server Build Beta 1
User: Alewis         Date:  2/26/99  Time:  5:04p
Labeled 'Server Build Beta 1'
Label comment:


**********************
Label: HostSDK 1.1
User: Cdunk          Date:  2/23/99  Time:  1:46p
Labeled 'HostSDK 1.1'
Label comment:
  Same as HostSDK 1.0 with a slightly different project
structure.

**********************
Label: Host SDK Install 1.0
User: Jsauer         Date:  2/15/99  Time: 11:44a
Labeled 'Host SDK Install 1.0'
Label comment:


**********************
Label: Server Snapshot for Descartes
User: Alewis         Date:  2/11/99  Time:  2:00p
```

Labeled 'Server Snapshot for Descartes'
Label comment:

****
Label: Build 1.5.0c
User: Alewis Date: 2/08/99 Time: 6:05p
Labeled 'Build 1.5.0c'
Label comment:

********** Version 177 **********
User: Alewis Date: 2/07/99 Time: 3:21p
Checked in $/Pagemail/Redirector
Comment:
changed log levels (mostly using LOG_OTHER for non-error events now that a different message code is used)

********** Version 176 **********
User: Alewis Date: 2/05/99 Time: 3:09p
Checked in $/Pagemail/redirector
Comment:
change where the "FromInboxOnly" is determined so it can be properly loaded from the user's mailbox configuration and acted upon

********** Version 175 **********
User: Alewis Date: 2/04/99 Time: 3:58p
Checked in $/Pagemail/redirector
Comment:
allow the server folder to be renamed using the registry; program wouldn't exit properly if SRP session could not be initialized

********** Version 174 **********
User: Alewis Date: 2/02/99 Time: 2:35p
Checked in $/Pagemail/Redirector
Comment:
logging refinements

****
Label: 1.5.0b
User: Alewis Date: 1/29/99 Time: 5:03p
Labeled '1.5.0b'
Label comment:

********** Version 173 **********
User: Alewis Date: 1/27/99 Time: 6:34p
Checked in $/Pagemail/Redirector
Comment:
add use of "pending" properties to smooth transitions from desktop to server; move some keys into LOCAL_MACHINE; support configurable number of worker threads

********** Version 172 **********
User: Alewis Date: 1/14/99 Time: 4:44p
Checked in $/Pagemail/Redirector
Comment:
allow specification to forward messages from all folders except deleted items

****
Label: Build 1.5.0a
User: Alewis Date: 1/12/99 Time: 4:52p

Labeled 'Build 1.5.0a'
Label comment:

********** Version 171 **********
User: Alewis Date: 1/11/99 Time: 10:41a
Checked in $/Pagemail/Redirector
Comment:
change Health checks to return boolean; periodically automatically check the "health" of the worker threads

****
Label: Build 1.5.0
User: Alewis Date: 1/04/99 Time: 2:32p
Labeled 'Build 1.5.0'
Label comment:

****
Label: Build 1.4.10
User: Alewis Date: 12/21/98 Time: 11:47a
Labeled 'Build 1.4.10'
Label comment:

****
Label: Build 1.4.9
User: Alewis Date: 12/16/98 Time: 2:37p
Labeled 'Build 1.4.9'
Label comment:

********** Version 170 **********
User: Alewis Date: 12/14/98 Time: 6:35p
Checked in $/Pagemail/Redirector
Comment:
better co-operation between desktop and server redirectors; server ONLY uses SRP now

****
Label: Build 1.0.8.1
User: Alewis Date: 12/11/98 Time: 5:02p
Labeled 'Build 1.0.8.1'
Label comment:

****
Label: Build 1.0.8
User: Alewis Date: 12/10/98 Time: 6:19p
Labeled 'Build 1.0.8'
Label comment:

********** Version 169 **********
User: Srahn Date: 12/10/98 Time: 1:17p
Checked in $/Pagemail/Redirector
Comment:
SRP code retrieves startup params from DB rather than registry.

****
Label: Build 1.0.7
User: Alewis Date: 12/07/98 Time: 5:45p
Labeled 'Build 1.0.7'

```
Label comment:


****************  Version 168  ****************
User: Alewis          Date: 12/04/98  Time:  7:25p
Checked in $/Pagemail/Redirector
Comment:
  make desktop version compile again

****************  Version 167  ****************
User: Vbanh           Date: 12/04/98  Time:  5:03p
Checked in $/Pagemail/Redirector
Comment:


**********************
Label: 1.0.7 Rogue 1
User: Alewis          Date: 12/04/98  Time:  4:56p
Labeled '1.0.7 Rogue 1'
Label comment:


****************  Version 166  ****************
User: Alewis          Date: 12/04/98  Time:  1:57p
Checked in $/Pagemail/redirector
Comment:
  change wording in some error messages

**********************
Label: pre-imap relay
User: Cdunk           Date: 12/02/98  Time:  2:07p
Labeled 'pre-imap relay'
Label comment:
  relay and the code it depends on prior to imap integration and
tertiary changes

****************  Version 165  ****************
User: Srahn           Date: 12/01/98  Time:  4:28p
Checked in $/Pagemail/redirector
Comment:
  SCS::StopHandheld() - call to PMDatabase::Delete() should not
attempt to remove server or client side databases.

****************  Version 164  ****************
User: Alewis          Date: 12/01/98  Time:  3:36p
Checked in $/Pagemail/redirector
Comment:
  no longer need to use user_key

****************  Version 163  ****************
User: Alewis          Date: 11/30/98  Time: 11:51a
Checked in $/Pagemail/redirector
Comment:
  new registry usage; change PocketLink to BlackBerry; support new
Pager database calls

****************  Version 162  ****************
User: Alewis          Date: 11/25/98  Time:  1:48p
Checked in $/Pagemail/redirector
Comment:
  server must recognize stats changes (since user may clear stats);
re-enable relay pinging; do INBOX rescan on a worker thread context
```

```
****************** Version 161 ******************
User: Alewis          Date: 11/24/98  Time:  1:41p
Checked in $/Pagemail/redirector
Comment:
  provide a "corrupt database" message for such cases

****************** Version 160 ******************
User: Alewis          Date: 11/23/98  Time:  4:08p
Checked in $/Pagemail/redirector
Comment:
  remove MAPI pinging

****************** Version 159 ******************
User: Alewis          Date: 11/23/98  Time:  8:10a
Checked in $/Pagemail/Redirector
Comment:
  remove dead code; clear some stats when starting; display result of
last transaction to device; perform filter first time message seen (rather than
just before sending)

************************
Label: Build 1.0.6
User: Alewis          Date: 11/17/98  Time:  3:44p
Labeled 'Build 1.0.6'
Label comment:


************************
Label: Build 1.0.5a
User: Alewis          Date: 11/16/98  Time:  5:23p
Labeled 'Build 1.0.5a'
Label comment:


****************** Version 158 ******************
User: Alewis          Date: 11/16/98  Time:  3:18p
Checked in $/Pagemail/Redirector
Comment:
  support for storing stats in MAPI database

************************
Label: Build 1.0.5
User: Alewis          Date: 11/12/98  Time:  2:40p
Labeled 'Build 1.0.5'
Label comment:


****************** Version 157 ******************
User: Srahn           Date: 11/11/98  Time:  4:09p
Checked in $/Pagemail/Redirector
Comment:
  Improved shutdown while still starting users.

************************
Label: Build 1.0.4
User: Alewis          Date: 11/11/98  Time:  1:55p
Labeled 'Build 1.0.4'
Label comment:


****************** Version 156 ******************
User: Srahn           Date: 11/10/98  Time:  9:37a
Checked in $/Pagemail/Redirector
```

```
Comment:
  Thread health feature.  Improved shutdown performance.

**********************
Label: Build 1.0.3
User: Alewis          Date: 11/09/98  Time:  6:17p
Labeled 'Build 1.0.3'
Label comment:


*****************  Version 155  *****************
User: Alewis          Date: 11/09/98  Time:  1:17p
Checked in $/Pagemail/redirector
Comment:
  modified logging; clean up warnings; initialize MAPI in threadproc;
combine two StartPager() routines into one

*****************  Version 154  *****************
User: Alewis          Date: 10/28/98  Time: 10:55a
Checked in $/Pagemail/Redirector
Comment:
  removal of dead code, and #ifdef'd code which will never be used
again

*****************  Version 153  *****************
User: Alewis          Date: 10/22/98  Time:  1:55p
Checked in $/Pagemail/Redirector
Comment:
  remove dependency on MAPI in the public interface for database

**********************
Label: Build 1.0.1
User: Alewis          Date: 10/19/98  Time:  5:28p
Labeled 'Build 1.0.1'
Label comment:


**********************
Label: Build 1.0.0
User: Alewis          Date: 10/16/98  Time:  6:18p
Labeled 'Build 1.0.0'
Label comment:


**********************
Label: Build 0.13.0
User: Alewis          Date: 10/09/98  Time:  1:59p
Labeled 'Build 0.13.0'
Label comment:


**********************
Label: Build 0.12.0
User: Alewis          Date: 10/07/98  Time:  3:45p
Labeled 'Build 0.12.0'
Label comment:


*****************  Version 152  *****************
User: Alewis          Date: 10/01/98  Time:  4:36p
Checked in $/Pagemail/redirector
Comment:
  send NO_DATABASE to dialog box if no pager could be started during
```

initialization

```
****************  Version 151  ****************
User: Roliver          Date: 10/01/98  Time:  3:15p
Checked in $/Pagemail/Redirector
Comment:
  Change the order of setting the pGlobalPager and calling
StartPager(), to hopefully have the pGP valid when the Dialog box tries to
update the stats...

**********************
Label: Build 0.11.7
User: Alewis           Date:  9/30/98  Time:  4:45p
Labeled 'Build 0.11.7'
Label comment:


**********************
Label: Build 0.11.6
User: Alewis           Date:  9/29/98  Time:  4:39p
Labeled 'Build 0.11.6'
Label comment:


**********************
Label: Build 0.11.5
User: Alewis           Date:  9/29/98  Time:  4:02p
Labeled 'Build 0.11.5'
Label comment:


**********************
Label: 0.11.5
User: Alewis           Date:  9/29/98  Time:  4:02p
Labeled '0.11.5'
Label comment:


**********************
Label: Build 0.11.4
User: Alewis           Date:  9/28/98  Time:  5:25p
Labeled 'Build 0.11.4'
Label comment:


**********************
Label: Build 0.11.0
User: Alewis           Date:  9/24/98  Time:  5:41p
Labeled 'Build 0.11.0'
Label comment:


****************  Version 150  ****************
User: Alewis           Date:  9/24/98  Time:  4:28p
Checked in $/Pagemail/redirector
Comment:
  change some instances of 'PageMail' to 'PocketLink'; use common
global for registry access; change location of registry settings for PocketLink

****************  Version 149  ****************
User: Roliver          Date:  9/23/98  Time:  7:08p
Checked in $/Pagemail/Redirector
Comment:
```

Add a numMsgsFailedForward count.
Ignore the InCradle immediately after verifying the relay address when going to RedirectionOn - this might be removed.

```
*****************  Version 148  *****************
User: Jsenders        Date:  9/22/98  Time:  3:55p
Checked in $/Pagemail/Redirector
Comment:
  call to db Open modified

*****************  Version 147  *****************
User: Roliver         Date:  9/22/98  Time:  9:55a
Checked in $/Pagemail/Redirector
Comment:
  just change a printf

**********************
Label: Build 0.10.0
User: Alewis          Date:  9/21/98  Time:  2:51p
Labeled 'Build 0.10.0'
Label comment:
  The "real" one?

*****************  Version 146  *****************
User: Roliver         Date:  9/18/98  Time:  4:40p
Checked in $/Pagemail/Redirector
Comment:
  #ifdef out the stats for server version

*****************  Version 145  *****************
User: Roliver         Date:  9/16/98  Time: 11:23a
Checked in $/Pagemail/Redirector
Comment:
  Use a RAM copy of PagerStats only. Don't use the MAPI db

*****************  Version 144  *****************
User: Cdunk           Date:  9/14/98  Time: 12:48p
Checked in $/Pagemail/redirector
Comment:
  fix pointer initialization error, caused possible exceptions on
start user

*****************  Version 143  *****************
User: Jsenders        Date:  9/09/98  Time: 11:32a
Checked in $/Pagemail/Redirector
Comment:
  removed a bug in StopPager related to deleting a user.

**********************
Label: Build.008
User: Roliver         Date:  9/04/98  Time:  4:13p
Labeled 'Build.008'
Label comment:
  Next release of redirector for internal beta.

*****************  Version 142  *****************
User: Roliver         Date:  9/04/98  Time:  1:01p
Checked in $/Pagemail/Redirector
Comment:
  Open the database with a ServiceName, Server or Redirector.

*****************  Version 141  *****************
User: Roliver         Date:  9/03/98  Time:  1:03p
```

```
Checked in $/Pagemail/Redirector
Comment:
  delete the pPgr correctly in StopPager() [checked in for Jacob]

****************** Version 140 ******************
User: Roliver          Date:  9/01/98  Time: 10:49a
Checked in $/Pagemail/Redirector
Comment:
  Fix a memory leak.
Remove some addressed "asdf" comments.
Add a filteredmsgs count to OnStat()

****************** Version 139 ******************
User: Roliver          Date:  8/28/98  Time:  5:35p
Checked in $/Pagemail/Redirector
Comment:
  Change #ifdef PERSONAL to EMAIL_TRANSPORT and PERSONAL_BUILD

****************** Version 138 ******************
User: Jsenders         Date:  8/26/98  Time: 10:44a
Checked in $/Pagemail/PageMail
Comment:
  DB interface modification to allow user more control over creation
and deletion of client db component. Error codes added to inform users of the
client db's internal state.

****************** Version 137 ******************
User: Jsenders         Date:  8/24/98  Time: 12:23p
Checked in $/Pagemail/PageMail
Comment:
  bug fix to ensure that the destructors of the implementation objects
derived from SysInfo and PMDatabase interfaces are called

****************** Version 136 ******************
User: Jsenders         Date:  8/24/98  Time: 11:51a
Checked in $/Pagemail/PageMail
Comment:
  changes to remove a circular dependance between objects in
SCS::StartPager()

***********************
Label: Build.007
User: Roliver          Date:  8/20/98  Time:  4:29p
Labeled 'Build.007'
Label comment:
  True Release of Redirector/Relay/PMDB for Internal Beta.

****************** Version 135 ******************
User: Roliver          Date:  8/20/98  Time:  4:25p
Checked in $/Pagemail/PageMail
Comment:
  Fix a typo for corporate build.

***********************
Label: Build.006
User: Roliver          Date:  8/19/98  Time:  4:21p
Labeled 'Build.006'
Label comment:
  Release for Redirector Internal RIM Beta Test.

****************** Version 134 ******************
User: Roliver          Date:  8/19/98  Time: 11:40a
Checked in $/Pagemail/PageMail
```

```
Comment:
  Use proper call syntax for GetServerDN() (Corporate build only)

****************** Version 133 ******************
User: Jsenders        Date:  8/18/98  Time:  4:38p
Checked in $/Pagemail/Redirector
Comment:
  database reorganization to abstract exported interfaces from MAPI
dependencies and remove the number of exported db interface

****************** Version 132 ******************
User: Roliver         Date:  8/17/98  Time:  5:16p
Checked in $/Pagemail/PageMail
Comment:
  Set the datagram ports to 0 for now (in server version).

**********************
Label: Build.005
User: Alewis          Date:  8/14/98  Time:  3:38p
Labeled 'Build.005'
Label comment:
  (Build .004 was never officially labelled)

****************** Version 131 ******************
User: Roliver         Date:  8/11/98  Time:  9:02p
Checked in $/Pagemail/Redirector
Comment:
  Add ExpiredMsg to OnStat() and pass EnableMapiPing to UserControl

****************** Version 130 ******************
User: Alewis          Date:  8/07/98  Time: 10:24a
Checked in $/Pagemail/Redirector
Comment:
  New calling convention for Device::SubsystemInitialize

****************** Version 129 ******************
User: Jsenders        Date:  8/06/98  Time:  4:49p
Checked in $/Pagemail/Redirector
Comment:
  db redesign and reorganization for server & personal PM

****************** Version 128 ******************
User: Jsenders        Date:  8/04/98  Time: 11:04a
Checked in $/Pagemail/Redirector
Comment:


****************** Version 127 ******************
User: Jsenders        Date:  8/04/98  Time: 10:49a
Checked in $/Pagemail/Redirector
Comment:


****************** Version 126 ******************
User: Jsenders        Date:  8/04/98  Time: 10:04a
Checked in $/Pagemail/Redirector
Comment:
  database redesign & new implementation for server PM and personal PM

****************** Version 125 ******************
User: Roliver         Date:  7/29/98  Time:  4:05p
Checked in $/Pagemail/Redirector
Comment:
```

modify some debuglog levels

```
*****************  Version 124  *****************
User: Alewis          Date:  7/29/98  Time:  3:46p
Checked in $/Pagemail/Redirector
Comment:
  Conform to new interface to debug log

***********************
Label: Build.003
User: Alewis          Date:  7/27/98  Time:  3:27p
Labeled 'Build.003'
Label comment:


*****************  Version 123  *****************
User: Alewis          Date:  7/27/98  Time:  2:15p
Checked in $/Pagemail/Redirector
Comment:
  Downgrade some LOG_INFORMATIONAL messages to LOG_DEBUG

***********************
Label: Build.002
User: Alewis          Date:  7/24/98  Time:  5:03p
Labeled 'Build.002'
Label comment:


*****************  Version 122  *****************
User: Alewis          Date:  7/23/98  Time:  4:14p
Checked in $/Pagemail/Redirector
Comment:
  Mailbox system must be closed after the database is closed

*****************  Version 121  *****************
User: Roliver         Date:  7/23/98  Time:  9:56a
Checked in $/Pagemail/Redirector
Comment:
  Change some Debug Levels in Printfs

*****************  Version 120  *****************
User: Roliver         Date:  7/22/98  Time:  1:09p
Checked in $/Pagemail/Redirector
Comment:
  fix typo on a delete call

*****************  Version 119  *****************
User: Roliver         Date:  7/21/98  Time:  4:11p
Checked in $/Pagemail/Redirector
Comment:
  Support the ConfigStatus in OnStat()

*****************  Version 118  *****************
User: Alewis          Date:  7/20/98  Time: 11:17a
Checked in $/Pagemail/PageMail
Comment:
  Use STL instead of home-brew container class; remove alarm:start()
and stop()

*****************  Version 117  *****************
User: Jsenders        Date:  7/17/98  Time: 10:56a
Checked in $/Pagemail/Redirector
Comment:
```

changes in handling char string properties

********** Version 116 **********
User: Roliver Date: 7/16/98 Time: 6:31p
Checked in $/Pagemail/PageMail
Comment:
Open the database with a parameter based on #ifdef PERSONAL

********** Version 115 **********
User: Roliver Date: 7/16/98 Time: 2:21p
Checked in $/Pagemail/PageMail
Comment:
Provide a global pointer to the one CPager (owned by scs) to the RedirectorDlg [same strategy as for the scs pointer].

********** Version 114 **********
User: Alewis Date: 7/13/98 Time: 2:00p
Checked in $/Pagemail/Redirector
Comment:
Changed HashList to take the size of the hash index as a parameter

********** Version 113 **********
User: Jsenders Date: 7/10/98 Time: 11:12a
Checked in $/Pagemail/Redirector
Comment:
fixed bug which would cause PPM to crash when more than 1 message exist in the Pagers folder.

****
Label: Build.001
User: Alewis Date: 7/09/98 Time: 5:06p
Labeled 'Build.001'
Label comment:
First official build. Lock and Load!

********** Version 112 **********
User: Roliver Date: 7/09/98 Time: 12:21p
Checked in $/Pagemail/PageMail
Comment:
Use the CURRENT_USER registry key group.

********** Version 111 **********
User: Roliver Date: 7/07/98 Time: 10:51p
Checked in $/Pagemail/Redirector
Comment:
Check for a possible null pointer before dereferencing...

********** Version 110 **********
User: Jsenders Date: 7/06/98 Time: 5:33p
Checked in $/Pagemail/Redirector

********** Version 109 **********
User: Jsenders Date: 7/06/98 Time: 4:53p
Checked in $/Pagemail/Redirector

********** Version 108 **********
User: Roliver Date: 7/03/98 Time: 4:18p
Checked in $/Pagemail/Redirector
Comment:
Send an error message to the caller of InitiliazeSystem().

********** Version 107 **********
User: Roliver Date: 6/26/98 Time: 2:28p

Checked in $/Pagemail/PageMail
Comment:
  Take out RelayDomainName read from Registry.
Call NotifyDatabaseChange() for a given userControl, when the database has been modified.

********** Version 106 **********
User: Roliver Date: 6/19/98 Time: 10:46a
Checked in $/Pagemail/PageMail
Comment:
  just rename to RelayDomainName

********** Version 105 **********
User: Srahn Date: 6/18/98 Time: 4:29p
Checked in $/Pagemail/PageMail
Comment:
  Get MAPI profile from the registry

********** Version 104 **********
User: Roliver Date: 6/18/98 Time: 1:48p
Checked in $/Pagemail/PageMail
Comment:
  Hardcode the redirection state to true for now.
Remove the RelayEmailAccount from Usercontrol/Registry

********** Version 103 **********
User: Alewis Date: 6/18/98 Time: 10:23a
Checked in $/Pagemail/PageMail
Comment:
  Changed the interface from "device" to "transport". The transport layer is now represented by a single object. Individual device objects have been moved into the PageMail project.

********** Version 102 **********
User: Roliver Date: 6/17/98 Time: 5:56p
Checked in $/Pagemail/PageMail
Comment:
  Don't create a format object for UserControl.
Create a filter object for UserControl.
Pass both RelayPagemailUser and RelayEmailAccount to UserControl from Registry.

********** Version 101 **********
User: Jsenders Date: 6/17/98 Time: 12:17p
Checked in $/Pagemail/PageMail

********** Version 100 **********
User: Jsenders Date: 6/17/98 Time: 10:03a
Checked in $/Pagemail/PageMail

********** Version 99 **********
User: Jsenders Date: 6/11/98 Time: 9:21a
Checked in $/Pagemail/PageMail

********** Version 98 **********
User: Roliver Date: 6/10/98 Time: 6:56p
Checked in $/Pagemail/PageMail
Comment:
  fix reg bug

********** Version 97 **********
User: Roliver Date: 6/10/98 Time: 6:36p
Checked in $/Pagemail/PageMail
Comment:

```
  add non-stall indicator...

****************  Version 96  ****************
User: Roliver        Date:  6/10/98  Time:  5:56p
Checked in $/Pagemail/PageMail
Comment:
  change __PERSONAL__ to PERSONAL, and fix ifdef build bug

****************  Version 95  ****************
User: Jsenders       Date:  6/10/98  Time:  5:41p
Checked in $/Pagemail/PageMail
Comment:
  adding filtering interface to Profile object

****************  Version 94  ****************
User: Jsenders       Date:  6/10/98  Time: 12:23p
Checked in $/Pagemail/PageMail
Comment:
  modifications to StartPager

****************  Version 93  ****************
User: Jsenders       Date:  6/10/98  Time: 11:51a
Checked in $/Pagemail/PageMail

****************  Version 92  ****************
User: Jsenders       Date:  6/09/98  Time:  5:06p
Checked in $/Pagemail/PageMail
Comment:
  New database module intergration,
PageMail modification to reflect new database model.

****************  Version 91  ****************
User: Cdunk          Date:  6/03/98  Time:  1:23p
Checked in $/Pagemail/PageMail

****************  Version 90  ****************
User: Cdunk          Date:  6/03/98  Time:  1:03p
Checked in $/Pagemail/PageMail

****************  Version 89  ****************
User: Alewis         Date:  6/02/98  Time: 12:34p
Checked in $/Pagemail/PageMail
Comment:
  Added a mutex around access to member variable within the global
notification class objects (pCurrentUser and pSystemInfo)

**********************
Label: 0.0.4
User: Roliver        Date:  6/01/98  Time: 11:46a
Labeled '0.0.4'
Label comment:
  This label is pre- Personal Pagemail. GME, CMIME and ETP have
been added to the project, though not fully incorporated into usercontrol as of
this label.

****************  Version 88  ****************
User: Roliver        Date:  5/05/98  Time: 10:53a
Checked in $/Pagemail/PageMail
Comment:
  Only SaveAndUnlock() when LoadAndLock() succeeded.

****************  Version 87  ****************
User: Roliver        Date:  5/05/98  Time: 10:16a
```

```
Checked in $/Pagemail/PageMail
Comment:
  change printf level

****************** Version 86 ******************
User: Roliver          Date:  5/05/98  Time:  9:53a
Checked in $/Pagemail/PageMail
Comment:
  printf mods

****************** Version 85 ******************
User: Roliver          Date:  5/05/98  Time:  9:31a
Checked in $/Pagemail/PageMail
Comment:
  Add a printf

****************** Version 84 ******************
User: Ipatters         Date:  4/29/98  Time:  3:23p
Checked in $/Pagemail/PageMail
Comment:
  Add printf for shutting down each user.

****************** Version 83 ******************
User: Roliver          Date:  4/28/98  Time:  1:06p
Checked in $/Pagemail/PageMail
Comment:
  change printf level again!

****************** Version 82 ******************
User: Roliver          Date:  4/28/98  Time:  1:04p
Checked in $/Pagemail/PageMail
Comment:
  change a printf level

****************** Version 81 ******************
User: Roliver          Date:  4/28/98  Time: 12:28p
Checked in $/Pagemail/PageMail
Comment:
  Change printf level

****************** Version 80 ******************
User: Roliver          Date:  4/28/98  Time: 12:01p
Checked in $/Pagemail/PageMail
Comment:
  Add a printf

****************** Version 79 ******************
User: Roliver          Date:  4/27/98  Time:  5:17p
Checked in $/Pagemail/PageMail
Comment:
  Add Alarm::Start() and Stop()

**********************
Label: 0.0.2
User: Roliver          Date:  4/24/98  Time:  3:28p
Labeled '0.0.2'
Label comment:
  Add read/delete/moved notification support - pending
DeviceSends are now cancelled on this notifications.
Handle DeviceSend responses: Expired -> immediate retry, RejectedByNetwork ->
slow retry, RejectedByDevice & RejectedByPacketblaster -> cancel;
Includes Allans changes to the Utilities, after the code review.
```

```
****************** Version 78 ******************
User: Alewis          Date:  4/23/98  Time:  5:34p
Checked in $/Pagemail/PageMail
Comment:
  Update for changes in Thread.h

****************** Version 77 ******************
User: Alewis          Date:  4/22/98  Time: 11:10a
Checked in $/Pagemail/PageMail

****************** Version 76 ******************
User: Roliver         Date:  4/21/98  Time:  2:00p
Checked in $/Pagemail/PageMail
Comment:
  Check the stop event after each new user is Started.

***********************
Label: Version 0.0.1
User: Roliver         Date:  4/21/98  Time:  1:59p
Labeled 'Version 0.0.1'
Label comment:
  Uses a Worker thread pool.
EntryId's are used to track new mail.
( Delete and Read cancelling is not yet implemented )
( Send/Receive failures are not yet handled )

****************** Version 75 ******************
User: Alewis          Date:  4/16/98  Time:  4:32p
Checked in $/Pagemail/PageMail

****************** Version 74 ******************
User: Alewis          Date:  3/30/98  Time: 10:19a
Checked in $/Pagemail/PageMail

****************** Version 73 ******************
User: Alewis          Date:  3/27/98  Time:  4:51p
Checked in $/Pagemail/PageMail

****************** Version 72 ******************
User: Roliver         Date:  3/27/98  Time: 11:22a
Checked in $/Pagemail/PageMail
Comment:
  Change over to a WorkerThread pool model. UserControl no longer
derives from Thread, the ThreadProc is effectively replaced with the DoWork()
method.

****************** Version 71 ******************
User: Alewis          Date:  3/20/98  Time:  4:20p
Checked in $/Pagemail/PageMail

****************** Version 70 ******************
User: Alewis          Date:  3/12/98  Time: 11:20a
Checked in $/Pagemail/PageMail

****************** Version 69 ******************
User: Alewis          Date:  3/11/98  Time: 11:30a
Checked in $/Pagemail/PageMail

****************** Version 68 ******************
User: Alewis          Date:  3/10/98  Time:  3:42p
Checked in $/Pagemail/PageMail
```

********** Version 67 **********
User: Alewis Date: 3/09/98 Time: 5:34p
Checked in $/Pagemail/PageMail

********** Version 66 **********
User: Alewis Date: 3/09/98 Time: 1:31p
Checked in $/Pagemail/PageMail

********** Version 65 **********
User: Alewis Date: 3/09/98 Time: 10:01a
Checked in $/Pagemail/PageMail

********** Version 64 **********
User: Alewis Date: 3/06/98 Time: 4:49p
Checked in $/Pagemail/PageMail

********** Version 63 **********
User: Pagemailadmin Date: 3/03/98 Time: 4:54p
Checked in $/Pagemail/PageMail

********** Version 62 **********
User: Alewis Date: 3/03/98 Time: 9:10a
Checked in $/Pagemail/PageMail

********** Version 61 **********
User: Alewis Date: 3/02/98 Time: 7:55a
Checked in $/Pagemail/PageMail

********** Version 60 **********
User: Alewis Date: 2/25/98 Time: 1:21p
Checked in $/Pagemail/PageMail

********** Version 59 **********
User: Alewis Date: 2/23/98 Time: 4:18p
Checked in $/Pagemail/PageMail

********** Version 58 **********
User: Alewis Date: 2/23/98 Time: 2:04p
Checked in $/Pagemail/PageMail

********** Version 57 **********
User: Alewis Date: 2/23/98 Time: 9:40a
Checked in $/Pagemail/PageMail

********** Version 56 **********
User: Alewis Date: 2/19/98 Time: 1:02p
Checked in $/Pagemail/PageMail

********** Version 55 **********
User: Alewis Date: 2/19/98 Time: 11:29a
Checked in $/Pagemail/PageMail

********** Version 54 **********
User: Alewis Date: 2/19/98 Time: 11:20a
Checked in $/Pagemail/PageMail

********** Version 53 **********
User: Alewis Date: 2/19/98 Time: 10:57a
Checked in $/Pagemail/PageMail

********** Version 52 **********
User: Alewis Date: 2/19/98 Time: 10:53a
Checked in $/Pagemail/PageMail

```
****************** Version 51 ******************
User: Alewis          Date:  2/18/98  Time:  2:07p
Checked in $/Pagemail/PageMail

****************** Version 50 ******************
User: Alewis          Date:  2/18/98  Time:  1:42p
Checked in $/Pagemail/PageMail

****************** Version 49 ******************
User: Alewis          Date:  2/17/98  Time: 12:50p
Checked in $/Pagemail/PageMail

****************** Version 48 ******************
User: Alewis          Date:  2/17/98  Time: 10:19a
Checked in $/Pagemail/PageMail

****************** Version 47 ******************
User: Alewis          Date:  2/16/98  Time:  4:42p
Checked in $/Pagemail/PMServe

****************** Version 46 ******************
User: Alewis          Date:  2/09/98  Time:  5:30p
Checked in $/Pagemail/Server

****************** Version 45 ******************
User: Alewis          Date:  2/06/98  Time:  4:58p
Checked in $/Pagemail/Server

****************** Version 44 ******************
User: Srahn           Date:  2/05/98  Time:  3:40p
Checked in $/Pagemail/Server

****************** Version 43 ******************
User: Srahn           Date:  2/05/98  Time:  1:49p
Checked in $/Pagemail/Server

****************** Version 42 ******************
User: Alewis          Date:  2/03/98  Time: 10:11a
Checked in $/Pagemail/Server

****************** Version 41 ******************
User: Alewis          Date:  1/29/98  Time:  5:38p
Checked in $/Pagemail/Server

****************** Version 40 ******************
User: Alewis          Date:  1/21/98  Time:  2:14p
Checked in $/Pagemail/Server

****************** Version 39 ******************
User: Alewis          Date:  1/13/98  Time:  5:32p
Checked in $/Pagemail/Server

****************** Version 38 ******************
User: Pagemailadmin   Date: 11/19/97  Time:  7:15p
Checked in $/Server

****************** Version 37 ******************
User: Cdunk           Date: 11/17/97  Time:  6:07p
Checked in $/Server

****************** Version 36 ******************
User: Cdunk           Date: 11/14/97  Time: 11:27a
```

Checked in $/Server

********** Version 35 **********
User: Pagemailadmin Date: 11/10/97 Time: 10:03a
Checked in $/Server

********** Version 34 **********
User: Cdunk Date: 11/07/97 Time: 2:42p
Checked in $/Server

********** Version 33 **********
User: Pagemailadmin Date: 11/06/97 Time: 5:52p
Checked in $/Server

********** Version 32 **********
User: Cdunk Date: 11/06/97 Time: 5:08p
Checked in $/Server

********** Version 31 **********
User: Pagemailadmin Date: 10/31/97 Time: 2:52p
Checked in $/Server

********** Version 30 **********
User: Cdunk Date: 10/31/97 Time: 9:07a
Checked in $/Server

********** Version 29 **********
User: Cdunk Date: 10/30/97 Time: 4:19p
Checked in $/Server

********** Version 28 **********
User: Pagemailadmin Date: 10/30/97 Time: 9:05a
Checked in $/Server

********** Version 27 **********
User: Cdunk Date: 10/29/97 Time: 3:00p
Checked in $/Server

********** Version 26 **********
User: Pagemailadmin Date: 10/28/97 Time: 11:18a
Checked in $/Server

********** Version 25 **********
User: Pagemailadmin Date: 10/24/97 Time: 4:23p
Checked in $/Server

********** Version 24 **********
User: Pagemailadmin Date: 10/24/97 Time: 2:42p
Checked in $/Server

********** Version 23 **********
User: Pagemailadmin Date: 10/23/97 Time: 4:40p
Checked in $/Server

********** Version 22 **********
User: Pagemailadmin Date: 10/16/97 Time: 12:55p
Checked in $/Server

********** Version 21 **********
User: Pagemailadmin Date: 10/16/97 Time: 9:32a
Checked in $/Server

********** Version 20 **********

User: Pagemailadmin Date: 10/15/97 Time: 12:56p
Checked in $/Server

********** Version 19 **********
User: Mbrown Date: 10/10/97 Time: 10:14a
Checked in $/Server

********** Version 18 **********
User: Mbrown Date: 10/10/97 Time: 9:28a
Checked in $/Server

********** Version 17 **********
User: Mbrown Date: 10/10/97 Time: 9:27a
Checked in $/Server

********** Version 16 **********
User: Mbrown Date: 10/10/97 Time: 9:21a
Checked in $/Server

********** Version 15 **********
User: Cdunk Date: 10/09/97 Time: 1:18p
Checked in $/Server

********** Version 14 **********
User: Bgilhuly Date: 10/09/97 Time: 9:58a
Checked in $/Server

********** Version 13 **********
User: Bgilhuly Date: 10/09/97 Time: 9:27a
Checked in $/Server

********** Version 12 **********
User: Cdunk Date: 10/08/97 Time: 9:31a
Checked in $/Server

********** Version 11 **********
User: Bgilhuly Date: 10/06/97 Time: 4:19p
Checked in $/Server

********** Version 10 **********
User: Cdunk Date: 10/06/97 Time: 1:27p
Checked in $/Server

********** Version 9 **********
User: Bgilhuly Date: 10/03/97 Time: 3:46p
Checked in $/Server
Comment:
  ready to integrate

********** Version 8 **********
User: Bgilhuly Date: 10/02/97 Time: 6:12p
Checked in $/Server

********** Version 7 **********
User: Nomad Date: 9/30/97 Time: 10:31a
Checked in $/Server

********** Version 6 **********
User: Bgilhuly Date: 9/29/97 Time: 10:42a
Checked in $/Server

********** Version 5 **********
User: Cdunk Date: 9/25/97 Time: 3:41p

```
Checked in $/Server

*****************  Version 4  *****************
User: Cdunk           Date:  9/25/97  Time:  3:22p
Checked in $/Server

*****************  Version 3  *****************
User: Cdunk           Date:  9/25/97  Time:  3:12p
Checked in $/Server

*****************  Version 2  *****************
User: Cdunk           Date:  9/25/97  Time:  2:28p
Checked in $/Server

*****************  Version 1  *****************
User: Cdunk           Date:  9/25/97  Time:  1:20p
Created scs.cpp
```

$/Pagemail/Redirector/usercontrol.cpp

****
Label: Build 1.6 RC1b
User: Alewis Date: 5/18/99 Time: 1:01p
Labeled 'Build 1.6 RC1b'
Label comment:
adds auto-signature

****
Label: Build 1.6 RC1a
User: Alewis Date: 5/14/99 Time: 5:40p
Labeled 'Build 1.6 RC1a'
Label comment:

********** Version 312 **********
User: Alewis Date: 5/14/99 Time: 12:54p
Checked in $/Pagemail/Redirector
Comment:
filter.h defines different constants for initial and "more" defaults

********** Version 311 **********
User: Alewis Date: 5/11/99 Time: 1:59p
Checked in $/Pagemail/Redirector
Comment:
initialize string in case a device ID cannot be retrieved

****
Label: Build 1.6 RC1
User: Alewis Date: 5/04/99 Time: 7:58p
Labeled 'Build 1.6 RC1'
Label comment:

********** Version 310 **********
User: Alewis Date: 5/03/99 Time: 11:33a
Checked in $/Pagemail/Redirector
Comment:
some extra logging

****
Label: Build 1.6.0 Desktop
User: Alewis Date: 4/30/99 Time: 12:31a
Labeled 'Build 1.6.0 Desktop'
Label comment:

****
Label: Checkpoint pre-RC1 Server
User: Alewis Date: 4/29/99 Time: 1:03a
Labeled 'Checkpoint pre-RC1 Server'
Label comment:

********** Version 309 **********
User: Alewis Date: 4/27/99 Time: 12:32p
Checked in $/Pagemail/redirector
Comment:
a little extra logging so we can see what size the GME blocks are
when sending messages to users

********** Version 308 **********

User: Alewis Date: 4/21/99 Time: 10:09a
Checked in $/Pagemail/Redirector
Comment:
remove dead source line

***************** Version 307 *****************
User: Tferguson Date: 4/08/99 Time: 11:04a
Checked in $/Pagemail/Redirector
Comment:
Changed LOG_ERROR to LOG_WARNING in HandleDatabaseChange() call to GetDeviceID().

****
Label: Checkpoint April 5
User: Alewis Date: 4/05/99 Time: 6:19p
Labeled 'Checkpoint April 5'
Label comment:

***************** Version 306 *****************
User: Alewis Date: 4/01/99 Time: 8:33p
Checked in $/Pagemail/Redirector
Comment:
account for nul at end of server name

***************** Version 305 *****************
User: Gvuong Date: 4/01/99 Time: 5:14p
Checked in $/Pagemail/Redirector
Comment:
Moved the GetMessageId function so that the reply status property can be set when there is no original message

****
Label: Server Build Beta 2b
User: Alewis Date: 4/01/99 Time: 2:59a
Labeled 'Server Build Beta 2b'
Label comment:

***************** Version 304 *****************
User: Alewis Date: 4/01/99 Time: 1:08a
Checked in $/Pagemail/redirector
Comment:
support separate stats and config notifications

****
Label: Server Build Beta 2a
User: Alewis Date: 3/26/99 Time: 11:04a
Labeled 'Server Build Beta 2a'
Label comment:

****
Label: Server Build Beta 2
User: Alewis Date: 3/24/99 Time: 6:16p
Labeled 'Server Build Beta 2'
Label comment:

***************** Version 303 *****************
User: Cdunk Date: 3/22/99 Time: 10:49a
Checked in $/Pagemail/Redirector
Comment:

Changes required for new database implementation

```
*****************  Version 302  *****************
User: Cdunk           Date:  3/21/99  Time:  3:26p
Checked in $/Pagemail/Redirector
Comment:
  Changes to make use of new database interface

***********************
Label: Database Change - Interim Label
User: Cdunk           Date:  3/21/99  Time:  3:09p
Labeled 'Database Change - Interim Label'
Label comment:
  Changes to database are made, integrated into all the projects
in the server DSW, but not the database utilities or the pocketlink projects.
Get this label to build the pocketlink or database utility projects.

*****************  Version 301  *****************
User: Alewis          Date:  3/12/99  Time:  7:13p
Checked in $/Pagemail/redirector
Comment:
  reload user database periodically in case notifications lost

*****************  Version 300  *****************
User: Alewis          Date:  3/08/99  Time:  8:34a
Checked in $/Pagemail/Redirector
Comment:
  remove unused variables

***********************
Label: Server Build Beta 1a
User: Alewis          Date:  3/05/99  Time:  2:01p
Labeled 'Server Build Beta 1a'
Label comment:


*****************  Version 299  *****************
User: Alewis          Date:  3/02/99  Time:  7:56a
Checked in $/Pagemail/Redirector
Comment:
  support SRP Cancel

*****************  Version 298  *****************
User: Alewis          Date:  3/02/99  Time:  7:52a
Checked in $/Pagemail/Redirector
Comment:
  remove dependence on windows.h

*****************  Version 297  *****************
User: Alewis          Date:  2/28/99  Time:  2:16p
Checked in $/Pagemail/Redirector
Comment:
  defer acknowledgment until after the message has been processed,
that was the relay won't think everything is good when there is a chance that
the message gets lost (due to the infamous MAPI hang)

***********************
Label: Server Build Beta 1
User: Alewis          Date:  2/26/99  Time:  5:04p
Labeled 'Server Build Beta 1'
Label comment:
```

****
Label: HostSDK 1.1
User: Cdunk Date: 2/23/99 Time: 1:46p
Labeled 'HostSDK 1.1'
Label comment:
Same as HostSDK 1.0 with a slightly different project structure.

****
Label: Host SDK Install 1.0
User: Jsauer Date: 2/15/99 Time: 11:44a
Labeled 'Host SDK Install 1.0'
Label comment:

****
Label: Server Snapshot for Descartes
User: Alewis Date: 2/11/99 Time: 2:00p
Labeled 'Server Snapshot for Descartes'
Label comment:

********** Version 296 **********
User: Alewis Date: 2/10/99 Time: 7:55p
Checked in $/Pagemail/redirector
Comment:
The GetNewMessages is not that critical. If there is a retryable error, we can wait until the next scan interval

****
Label: Build 1.5.0c
User: Alewis Date: 2/08/99 Time: 6:05p
Labeled 'Build 1.5.0c'
Label comment:

********** Version 295 **********
User: Alewis Date: 2/08/99 Time: 6:03p
Checked in $/Pagemail/redirector
Comment:
quietly change relay1@phoad.rim.net if found in the user's registry

********** Version 294 **********
User: Alewis Date: 2/08/99 Time: 3:58p
Checked in $/Pagemail/redirector
Comment:
allow the mailbox configuration to override the registry setting

********** Version 293 **********
User: Alewis Date: 2/07/99 Time: 3:19p
Checked in $/Pagemail/Redirector
Comment:
oops, wrong default for bRedirectFromAllFolders

********** Version 292 **********
User: Alewis Date: 2/05/99 Time: 5:34p
Checked in $/Pagemail/redirector
Comment:
non-compliant GME implementation - it was supposed to skip unrecognized GME headers

********** Version 291 **********
User: Alewis Date: 2/05/99 Time: 3:17p

```
Checked in $/Pagemail/redirector
Comment:
  switch bFromInboxOnly to bRedirectFromAllFolders

*****************  Version 290  *****************
User: Alewis          Date:  2/05/99  Time:  3:09p
Checked in $/Pagemail/redirector
Comment:
  change where the "FromInboxOnly" is determined so it can be properly
loaded from the user's mailbox configuration and acted upon

*****************  Version 289  *****************
User: Alewis          Date:  2/04/99  Time:  9:12a
Checked in $/Pagemail/Redirector
Comment:
  used the wrong stream to get the actual length

*****************  Version 288  *****************
User: Alewis          Date:  2/03/99  Time:  5:10p
Checked in $/Pagemail/redirector
Comment:
  if there is less than 1/4 the requested amount remaining in the
message, include it in the forward operation

*****************  Version 287  *****************
User: Alewis          Date:  2/02/99  Time:  2:35p
Checked in $/Pagemail/Redirector
Comment:
  logging refinements

***********************
Label: 1.5.0b
User: Alewis          Date:  1/29/99  Time:  5:03p
Labeled '1.5.0b'
Label comment:


*****************  Version 286  *****************
User: Alewis          Date:  1/28/99  Time:  7:23p
Checked in $/Pagemail/redirector
Comment:
  server keeps certain keys in LOCAL_MACHINE while desktop keeps them
in CURRENT_USER; clear pending count upon startup, even in server mode.

*****************  Version 285  *****************
User: Alewis          Date:  1/27/99  Time:  6:34p
Checked in $/Pagemail/Redirector
Comment:
  add use of "pending" properties to smooth transitions from desktop
to server; move some keys into LOCAL_MACHINE; support configurable number of
worker threads

*****************  Version 284  *****************
User: Alewis          Date:  1/18/99  Time:  6:04p
Checked in $/Pagemail/redirector
Comment:
  provide property to be used to disable delivery confirmations

*****************  Version 283  *****************
User: Alewis          Date:  1/18/99  Time:  9:04a
Checked in $/Pagemail/Redirector
Comment:
  adjustments to error reporting
```

```
****************** Version 282 ******************
User: Alewis          Date:  1/15/99  Time:  5:26p
Checked in $/Pagemail/redirector
Comment:
  support new definition of mailbox::GetLastError

****************** Version 281 ******************
User: Alewis          Date:  1/14/99  Time:  4:44p
Checked in $/Pagemail/Redirector
Comment:
  adjust some logging levels

****************** Version 280 ******************
User: Alewis          Date:  1/13/99  Time:  4:29p
Checked in $/Pagemail/redirector
Comment:
  change logging levels for some messages

**********************
Label: Build 1.5.0a
User: Alewis          Date:  1/12/99  Time:  4:52p
Labeled 'Build 1.5.0a'
Label comment:


****************** Version 279 ******************
User: Alewis          Date:  1/12/99  Time:  4:48p
Checked in $/Pagemail/redirector
Comment:
  registry key wasn't being created if it didn't already exist

****************** Version 278 ******************
User: Alewis          Date:  1/08/99  Time: 12:48p
Checked in $/Pagemail/redirector
Comment:
  calculate routing information to be put into GME header in a way
that it will match what was given to the pager

**********************
Label: Build 1.5.0
User: Alewis          Date:  1/04/99  Time:  2:32p
Labeled 'Build 1.5.0'
Label comment:


**********************
Label: Build 1.4.10
User: Alewis          Date: 12/21/98  Time: 11:47a
Labeled 'Build 1.4.10'
Label comment:


****************** Version 277 ******************
User: Vbanh           Date: 12/17/98  Time: 11:47a
Checked in $/Pagemail/Redirector
Comment:
  changed status

**********************
Label: Build 1.4.9
User: Alewis          Date: 12/16/98  Time:  2:37p
Labeled 'Build 1.4.9'
```

Label comment:

********** Version 276 **********
User: Alewis Date: 12/15/98 Time: 9:58a
Checked in $/Pagemail/redirector
Comment:
removed setting of STOPPED status. There was going to be a race condition between removing the user from the server, the server actually stopping the user, and the desktop redirector starting the user, such that the displayed status in the redirector may be "Stopped" even though it should be "Running".

********** Version 275 **********
User: Alewis Date: 12/14/98 Time: 6:35p
Checked in $/Pagemail/Redirector
Comment:
better co-operation between desktop and server redirectors; server ONLY uses SRP now

********** Version 274 **********
User: Alewis Date: 12/11/98 Time: 5:37p
Checked in $/Pagemail/Redirector
Comment:
use global NextEtpId in a thread-safe manner

********** Version 273 **********
User: Alewis Date: 12/11/98 Time: 5:06p
Checked in $/Pagemail/Redirector
Comment:
global next ETP variable

****
Label: Build 1.0.8.1
User: Alewis Date: 12/11/98 Time: 5:02p
Labeled 'Build 1.0.8.1'
Label comment:

********** Version 272 **********
User: Alewis Date: 12/11/98 Time: 4:58p
Checked in $/Pagemail/redirector
Comment:
add debug statement

****
Label: Build 1.0.8
User: Alewis Date: 12/10/98 Time: 6:19p
Labeled 'Build 1.0.8'
Label comment:

********** Version 271 **********
User: Ipatters Date: 12/10/98 Time: 4:11p
Checked in $/Pagemail/Redirector
Comment:
bGotRelayAddress not used by SRP version

********** Version 270 **********
User: Alewis Date: 12/09/98 Time: 5:53p
Checked in $/Pagemail/redirector
Comment:
HandHeld ->Handheld

```
****************  Version 269  ****************
User: Alewis          Date: 12/08/98  Time: 12:23p
Checked in $/Pagemail/redirector
Comment:
  added some more debug statements

**********************
Label: Build 1.0.7
User: Alewis          Date: 12/07/98  Time:  5:45p
Labeled 'Build 1.0.7'
Label comment:


****************  Version 268  ****************
User: Alewis          Date: 12/07/98  Time:  4:19p
Checked in $/Pagemail/redirector
Comment:
  re-enable rescanning of inbox

****************  Version 267  ****************
User: Alewis          Date: 12/07/98  Time:  1:35p
Checked in $/Pagemail/Redirector
Comment:
  change default e-mail forwarding address

****************  Version 266  ****************
User: Alewis          Date: 12/07/98  Time: 10:28a
Checked in $/Pagemail/Redirector
Comment:
  wasn't reloading email address override properly; ETP not used when
USE_SRP

****************  Version 265  ****************
User: Alewis          Date: 12/04/98  Time:  7:26p
Checked in $/Pagemail/Redirector
Comment:
  get email override address from database instead of registry

****************  Version 264  ****************
User: Alewis          Date: 12/04/98  Time:  7:26p
Checked in $/Pagemail/Redirector
Comment:
  make desktop version compile again

****************  Version 263  ****************
User: Vbanh           Date: 12/04/98  Time:  5:03p
Checked in $/Pagemail/Redirector
Comment:


**********************
Label: 1.0.7 Rogue 1
User: Alewis          Date: 12/04/98  Time:  4:56p
Labeled '1.0.7 Rogue 1'
Label comment:


****************  Version 262  ****************
User: Alewis          Date: 12/04/98  Time:  1:57p
Checked in $/Pagemail/redirector
Comment:
  FirstStartTime registry setting no longer used
```

```
**********************
Label: pre-imap relay
User: Cdunk           Date: 12/02/98  Time:  2:07p
Labeled 'pre-imap relay'
Label comment:
  relay and the code it depends on prior to imap integration and
tertiary changes

*****************  Version 261  *****************
User: Alewis          Date: 12/02/98  Time: 12:35p
Checked in $/Pagemail/Redirector
Comment:
  add some debugging; use CURRENT_USER, even for server; adjust
m_redirectionState determination; more?

*****************  Version 260  *****************
User: Alewis          Date: 11/30/98  Time: 11:53a
Checked in $/Pagemail/Redirector
Comment:
  new registry usage; change PocketLink to BlackBerry; add critical
section around Find/queue new mail work request; move <confirm> to start of
subject line

*****************  Version 259  *****************
User: Alewis          Date: 11/27/98  Time:  1:27p
Checked in $/Pagemail/Redirector
Comment:
  add debugging statements; move in-cradle filter to same location as
regular filter handling

*****************  Version 258  *****************
User: Alewis          Date: 11/26/98  Time:  5:04p
Checked in $/Pagemail/redirector
Comment:
  reduce logging level of a message; clear pending count at startup -
it will be refreshed automatically

*****************  Version 257  *****************
User: Alewis          Date: 11/25/98  Time:  4:28p
Checked in $/Pagemail/redirector
Comment:
  couple of extra debug statements

*****************  Version 256  *****************
User: Alewis          Date: 11/25/98  Time:  2:47p
Checked in $/Pagemail/redirector
Comment:
  shouldn't have been clearing stats automatically on startup

*****************  Version 255  *****************
User: Alewis          Date: 11/25/98  Time:  1:48p
Checked in $/Pagemail/redirector
Comment:
  server must recognize stats changes (since user may clear stats);
re-enable relay pinging; do INBOX rescan on a worker thread context

*****************  Version 254  *****************
User: Alewis          Date: 11/23/98  Time:  4:09p
Checked in $/Pagemail/redirector
Comment:
  remove MAPI pinging; tweak timeout processing; check
DeviceSendTransaction array before queuing new messages
```

```
***************** Version 253 *****************
User: Alewis          Date: 11/23/98  Time: 11:58a
Checked in $/Pagemail/Redirector
Comment:
  duplicate RemoveDeviceSendTransaction() removed; post LastXPAction
for all forwarding failures (even resource and mailbox failures)

***************** Version 252 *****************
User: Arogobete       Date: 11/23/98  Time: 10:33a
Checked in $/Pagemail/Redirector
Comment:
  modified the Workee timer period and added a call to
FindAndQueueAllNewMail in the timer handler

added a function call in FindAndQueueAllNewMail to check first if an ENtryID is
already in the Mailbox2Device queue before adding it

***************** Version 251 *****************
User: Alewis          Date: 11/23/98  Time:  8:10a
Checked in $/Pagemail/Redirector
Comment:
  remove dead code; clear some stats when starting; display result of
last transaction to device; perform filter first time message seen (rather than
just before sending)

***************** Version 250 *****************
User: Alewis          Date: 11/20/98  Time:  4:32p
Checked in $/Pagemail/redirector
Comment:
  improved user interface when server in control of redirection

***********************
Label: Build 1.0.6
User: Alewis          Date: 11/17/98  Time:  3:44p
Labeled 'Build 1.0.6'
Label comment:


***************** Version 249 *****************
User: Alewis          Date: 11/17/98  Time:  3:41p
Checked in $/Pagemail/redirector
Comment:
  clear pending messages count in statistics on startup

***************** Version 248 *****************
User: Srahn           Date: 11/17/98  Time:  1:57p
Checked in $/Pagemail/Redirector
Comment:
  Add case GME::MESSAGE_FROM_SERVER to UserControl::HandleDataComand
for DEBUG purposes only.  Used in loopback testing in the lab.

***********************
Label: Build 1.0.5a
User: Alewis          Date: 11/16/98  Time:  5:23p
Labeled 'Build 1.0.5a'
Label comment:


***************** Version 247 *****************
User: Alewis          Date: 11/16/98  Time:  3:18p
Checked in $/Pagemail/Redirector
Comment:
```

support for storing stats in MAPI database

****
Label: Build 1.0.5
User: Alewis Date: 11/12/98 Time: 2:40p
Labeled 'Build 1.0.5'
Label comment:

********** Version 246 **********
User: Alewis Date: 11/12/98 Time: 2:07p
Checked in $/Pagemail/redirector
Comment:
use same transaction and etpid for 4-hour message retransmit

****
Label: Build 1.0.4
User: Alewis Date: 11/11/98 Time: 1:55p
Labeled 'Build 1.0.4'
Label comment:

********** Version 245 **********
User: Alewis Date: 11/11/98 Time: 1:54p
Checked in $/Pagemail/Redirector
Comment:
change default relay email address

********** Version 244 **********
User: Alewis Date: 11/10/98 Time: 6:50p
Checked in $/Pagemail/Redirector
Comment:
wasn't handling ETP messages correctly that did not have a data stream associated with it

********** Version 243 **********
User: Alewis Date: 11/10/98 Time: 11:03a
Checked in $/Pagemail/Redirector
Comment:
wasn't using the LOCAL_MACHINE key in server mode

****
Label: Build 1.0.3
User: Alewis Date: 11/09/98 Time: 6:17p
Labeled 'Build 1.0.3'
Label comment:

********** Version 242 **********
User: Alewis Date: 11/09/98 Time: 3:55p
Checked in $/Pagemail/Redirector
Comment:
use relay@rim.net for now

********** Version 241 **********
User: Alewis Date: 11/09/98 Time: 3:18p
Checked in $/Pagemail/Redirector
Comment:
GME version 2 enabled

********** Version 240 **********
User: Alewis Date: 11/09/98 Time: 1:38p
Checked in $/Pagemail/Redirector

```
Comment:
  quick update - a stream was not being allocated before being used!

****************  Version 239  ****************
User: Alewis          Date: 11/09/98  Time:  1:19p
Checked in $/Pagemail/Redirector
Comment:
  massive code changes to support new GME format; cancel messages
after 4 hours if deleted or read (rather than retrying); modified logging;
proper use of ETP and GME transaction IDs

****************  Version 238  ****************
User: Alewis          Date: 11/02/98  Time: 12:32p
Checked in $/Pagemail/Redirector
Comment:
  Some code reorganization in anticipation for implementing GME
changes

****************  Version 237  ****************
User: Alewis          Date: 10/30/98  Time:  1:10p
Checked in $/Pagemail/Redirector
Comment:
  send MORE results immediately rather than placing it at the end of a
queue.

****************  Version 236  ****************
User: Alewis          Date: 10/28/98  Time: 10:55a
Checked in $/Pagemail/Redirector
Comment:
  removal of dead code, and #ifdef'd code which will never be used
again

****************  Version 235  ****************
User: Alewis          Date: 10/26/98  Time: 11:13a
Checked in $/Pagemail/redirector
Comment:
  Change "Pager Delivery Confirmation" to "PocketLink Delivery
Confirmation"

**********************
Label: Build 1.0.1
User: Alewis          Date: 10/19/98  Time:  5:28p
Labeled 'Build 1.0.1'
Label comment:


****************  Version 234  ****************
User: Alewis          Date: 10/19/98  Time:  5:24p
Checked in $/Pagemail/redirector
Comment:
  set timestamp on MORE responses

**********************
Label: Build 1.0.0
User: Alewis          Date: 10/16/98  Time:  6:18p
Labeled 'Build 1.0.0'
Label comment:


****************  Version 233  ****************
User: Alewis          Date: 10/16/98  Time: 12:12p
Checked in $/Pagemail/redirector
Comment:
```

handle timeout/expired case where message no longer exists in mailbox

********** Version 232 **********
User: Alewis Date: 10/15/98 Time: 6:24p
Checked in $/Pagemail/redirector
Comment:
add status display for redirection being suspended for user activity (start of "screen-save" method of suspension); move handling for insertion of text and attachments into mailbox code

********** Version 231 **********
User: Alewis Date: 10/13/98 Time: 7:32p
Checked in $/Pagemail/Redirector
Comment:
add some random information into the generation of hashed message IDs; format changes to the delivery confirmation e-mail message

********** Version 230 **********
User: Alewis Date: 10/13/98 Time: 11:43a
Checked in $/Pagemail/redirector
Comment:
downgrade a message to informational. UserControl::MarkMessageAsTransferred(): failed is likely if the message was deleted prior to receiving the notification from relay.

****
Label: Build 0.13.0
User: Alewis Date: 10/09/98 Time: 1:59p
Labeled 'Build 0.13.0'
Label comment:

****
Label: Build 0.12.0
User: Alewis Date: 10/07/98 Time: 3:45p
Labeled 'Build 0.12.0'
Label comment:

********** Version 229 **********
User: Alewis Date: 10/06/98 Time: 5:59p
Checked in $/Pagemail/Redirector
Comment:
refid in DoDeviceReceiveProcessing() should be an int

********** Version 228 **********
User: Alewis Date: 10/06/98 Time: 5:41p
Checked in $/Pagemail/redirector
Comment:
synch up with error codes expected by pager

********** Version 227 **********
User: Alewis Date: 10/06/98 Time: 3:52p
Checked in $/Pagemail/redirector
Comment:
provide registry settings to configure maximum amounts to be forwarded (initially and for "more" responses)

********** Version 226 **********
User: Alewis Date: 10/05/98 Time: 12:48p
Checked in $/Pagemail/redirector
Comment:

Another change to the delivery confirmation message

********** Version 225 **********
User: Alewis Date: 10/05/98 Time: 10:42a
Checked in $/Pagemail/redirector
Comment:
Change message for delivery confirmations

********** Version 224 **********
User: Cdunk Date: 10/02/98 Time: 1:57p
Checked in $/Pagemail/redirector
Comment:
Specifies flags to the relay as a suggestion to try duplicate elimination based on etpid.

********** Version 223 **********
User: Cdunk Date: 10/02/98 Time: 10:26a
Checked in $/Pagemail/Redirector
Comment:
verified retry changes. occurs on the work thread now and occurs both after a time-out and upon a receiving a ping response.

********** Version 222 **********
User: Roliver Date: 10/01/98 Time: 3:41p
Checked in $/Pagemail/Redirector
Comment:
Moved the RETRY timeout code into the general DoWork() routine, and defined the new WRP type CHECK_FOR_RETRIES.
Also, the PING_RESPONSE will immediately send retries for any oustanding messages to the relay.

********** Version 221 **********
User: Cdunk Date: 9/30/98 Time: 6:26p
Checked in $/Pagemail/redirector

********** Version 220 **********
User: Cdunk Date: 9/30/98 Time: 5:52p
Checked in $/Pagemail/redirector
Comment:
Eliminated one situation where one thread could be writing and the other reading from a given piece of data if the timing was bad.

********** Version 219 **********
User: Cdunk Date: 9/30/98 Time: 5:43p
Checked in $/Pagemail/redirector
Comment:
Using timers totally incorrectly, fixed, however, #ifndef EmailTransferProtocol code is also wrong, the incorrect code was commented out, but not fixed (it may never be used)

****
Label: Build 0.11.7
User: Alewis Date: 9/30/98 Time: 4:45p
Labeled 'Build 0.11.7'
Label comment:

********** Version 218 **********
User: Alewis Date: 9/30/98 Time: 2:15p
Checked in $/Pagemail/redirector
Comment:
use SERVER_ORIGINATED_REF_ID_BIT instead of magic number

********** Version 217 **********
User: Cdunk Date: 9/30/98 Time: 11:20a
Checked in $/Pagemail/redirector
Comment:
retry pager message after four hours, every four hours for all time.

********** Version 216 **********
User: Cdunk Date: 9/29/98 Time: 6:05p
Checked in $/Pagemail/redirector
Comment:
Stub code for 4 hour failsafe retry timer. Needs completion

****
Label: Build 0.11.6
User: Alewis Date: 9/29/98 Time: 4:39p
Labeled 'Build 0.11.6'
Label comment:

********** Version 215 **********
User: Alewis Date: 9/29/98 Time: 4:32p
Checked in $/Pagemail/redirector
Comment:
didn't load the message Id from the file if it was already there
(local variable not initialized)

****
Label: Build 0.11.5
User: Alewis Date: 9/29/98 Time: 4:02p
Labeled 'Build 0.11.5'
Label comment:

****
Label: 0.11.5
User: Alewis Date: 9/29/98 Time: 4:02p
Labeled '0.11.5'
Label comment:

********** Version 214 **********
User: Alewis Date: 9/29/98 Time: 2:20p
Checked in $/Pagemail/Redirector
Comment:
memory leak removing message IDs from new mail

********** Version 213 **********
User: Alewis Date: 9/29/98 Time: 1:38p
Checked in $/Pagemail/redirector
Comment:
Only generate message reference IDs if there isn't already one
generated by us (top bit set)

****
Label: Build 0.11.4
User: Alewis Date: 9/28/98 Time: 5:25p
Labeled 'Build 0.11.4'
Label comment:

********** Version 212 **********
User: Srahn Date: 9/28/98 Time: 1:22p
Checked in $/Pagemail/redirector

```
Comment:
  Replaced the calls GetDesktopEmailAddress with Mailbox::WhoAmI

****************** Version 211 ******************
User: Roliver          Date:  9/25/98  Time:  5:32p
Checked in $/Pagemail/Redirector
Comment:
  Don't set the forwarding address in the registry until the ping
response has arrived.

****************** Version 210 ******************
User: Alewis           Date:  9/25/98  Time:  3:12p
Checked in $/Pagemail/Redirector
Comment:
  remove most ETP processing from mailbox - put it into usercontrol;
add explicit call to add a reference ID to a message

**********************
Label: Build 0.11.0
User: Alewis           Date:  9/24/98  Time:  5:41p
Labeled 'Build 0.11.0'
Label comment:


****************** Version 209 ******************
User: Roliver          Date:  9/24/98  Time:  5:00p
Checked in $/Pagemail/Redirector
Comment:
  fix bug! if etp send more fails, then and only then remove the entry
from the transactionlist.
NOTE: the bug was harmless on successes

****************** Version 208 ******************
User: Alewis           Date:  9/24/98  Time:  4:27p
Checked in $/Pagemail/Redirector
Comment:
  go back at most 3 days on startup to find mail to forward

****************** Version 207 ******************
User: Alewis           Date:  9/24/98  Time:  3:30p
Checked in $/Pagemail/Redirector
Comment:
  use RootKeyString from SCS; formatting fix in a debug statement

****************** Version 206 ******************
User: Roliver          Date:  9/24/98  Time: 11:59a
Checked in $/Pagemail/Redirector
Comment:
  Don't increment the failedMsgs count if the GetRIMMessage() fails,
as this might occur if the user moves or deletes the msg before it is sent.

****************** Version 205 ******************
User: Roliver          Date:  9/23/98  Time:  7:08p
Checked in $/Pagemail/Redirector
Comment:
  Add a numMsgsFailedForward count.
Ignore the InCradle immediately after verifying the relay address when going to
RedirectionOn - this might be removed.

****************** Version 204 ******************
User: Alewis           Date:  9/22/98  Time:  9:48a
Checked in $/Pagemail/redirector
Comment:
```

Remove spaces in header prefixes - with proportional fonts spaces are bad for setting columns

****
Label: Build 0.10.0
User: Alewis Date: 9/21/98 Time: 2:51p
Labeled 'Build 0.10.0'
Label comment:
The "real" one?

********** Version 203 **********
User: Roliver Date: 9/21/98 Time: 1:17p
Checked in $/Pagemail/Redirector
Comment:
Use the GetDesktopAddress() to set the m_pMailboxName.
Send this desktopAddress with every GME as the source address.

********** Version 202 **********
User: Roliver Date: 9/21/98 Time: 11:23a
Checked in $/Pagemail/Redirector
Comment:
Allow the Redirector to send messages which include the original text of a message originally created by the pager.

********** Version 201 **********
User: Alewis Date: 9/21/98 Time: 9:46a
Checked in $/Pagemail/Redirector
Comment:
Removed too many spaces :-)

********** Version 200 **********
User: Alewis Date: 9/21/98 Time: 9:41a
Checked in $/Pagemail/redirector
Comment:
Minor formatting change to delivery confirmation (remove some spaces)

********** Version 199 **********
User: Roliver Date: 9/18/98 Time: 3:54p
Checked in $/Pagemail/Redirector
Comment:
Add startup code for unread mail.

********** Version 198 **********
User: Hhind Date: 9/18/98 Time: 1:30p
Checked in $/Pagemail/Redirector
Comment:
Fix gme parm index typo

********** Version 197 **********
User: Roliver Date: 9/18/98 Time: 12:30p
Checked in $/Pagemail/Redirector
Comment:
Incorporate GME format changes for GME::NEW_MESSAGE and GME::MESSAGE_TO_SUBMIT
(this changes the lists of TRANSACTION and MESSSAGE_ERRORs for GME)

********** Version 196 **********
User: Alewis Date: 9/17/98 Time: 10:16p
Checked in $/Pagemail/redirector
Comment:
delete RIMMessage when finished with it in HandleMoreRequest

```
****************  Version 195  ****************
User: Roliver        Date:  9/17/98  Time:  3:46p
Checked in $/Pagemail/Redirector
Comment:
  Support ServiceName to disable the redirection.
Send the ETP status to the originating Relay address.

****************  Version 194  ****************
User: Roliver        Date:  9/16/98  Time:  3:58p
Checked in $/Pagemail/Redirector
Comment:
  fix pointer bug
finish off confirmation message stuff

****************  Version 193  ****************
User: Roliver        Date:  9/16/98  Time: 12:09p
Checked in $/Pagemail/Redirector
Comment:
  Add <confirm> support.
Do retries on mapi calls, if retryable error

****************  Version 192  ****************
User: Roliver        Date:  9/14/98  Time:  2:00p
Checked in $/Pagemail/Redirector
Comment:
  Just forward a default number of days previous unread mail at
startup - desktop and server version.
ifdef out ServiceName stuff for now, til Jacob is done

****************  Version 191  ****************
User: Roliver        Date:  9/08/98  Time:  4:44p
Checked in $/Pagemail/Redirector
Comment:
  In SERVER version, always forward old unread mail that is no older
than 3 days, at startup.

**********************
Label: Build.008
User: Roliver        Date:  9/04/98  Time:  4:13p
Labeled 'Build.008'
Label comment:
  Next release of redirector for internal beta.

****************  Version 190  ****************
User: Roliver        Date:  9/04/98  Time:  1:02p
Checked in $/Pagemail/Redirector
Comment:
  #ifdef out ServiceName code until Jacob returns.
Remove some asdfs.
Forward all old main when not using ETP.

****************  Version 189  ****************
User: Roliver        Date:  9/03/98  Time:  1:22p
Checked in $/Pagemail/Redirector
Comment:
  Get an actualLength from the MoreResult() function.

****************  Version 188  ****************
User: Roliver        Date:  9/03/98  Time: 12:31p
Checked in $/Pagemail/Redirector
Comment:
  Don't forward old mail in server, when dont forward flag is set!
```

```
****************** Version 187 ******************
User: Roliver          Date:  9/03/98  Time: 11:18a
Checked in $/Pagemail/Redirector
Comment:
  Change some more gme errors from transaction to message errors.
Modify a debuglog level.

****************** Version 186 ******************
User: Srahn            Date:  9/03/98  Time: 10:18a
Checked in $/Pagemail/Redirector
Comment:
  declare dvcId !?

****************** Version 185 ******************
User: Roliver          Date:  9/02/98  Time:  9:45p
Checked in $/Pagemail/Redirector
Comment:
  again remove some out of date comments

****************** Version 184 ******************
User: Roliver          Date:  9/02/98  Time:  9:32p
Checked in $/Pagemail/Redirector
Comment:
  send a more specific gme error code, when a gme command has a bad
format.

****************** Version 183 ******************
User: Roliver          Date:  9/02/98  Time:  8:31p
Checked in $/Pagemail/Redirector
Comment:
  Always log an error if cannot send GME error status back to device
on a failure.
Remove some asdf comments.
Handle new ETP status-es

****************** Version 182 ******************
User: Roliver          Date:  9/02/98  Time:  7:52p
Checked in $/Pagemail/Redirector
Comment:
  Activate MORE command handling.
In Desktop version, check to see if a Server owns the account, if so, don't
allow the Desktop version to redirect.

****************** Version 181 ******************
User: Roliver          Date:  9/01/98  Time: 10:51a
Checked in $/Pagemail/Redirector
Comment:
  add a filteredmsgs count
remove some unnecessary "asdf" comments

****************** Version 180 ******************
User: Roliver          Date:  8/28/98  Time:  6:04p
Checked in $/Pagemail/PageMail
Comment:
  change #ifdef PERSONAL to EMAIL_TRANSPORT and PERSONAL_BUILD.
modify DebugPrintf so that mailboxname is always printed.

****************** Version 179 ******************
User: Cdunk            Date:  8/28/98  Time:  2:34p
Checked in $/Pagemail/PageMail
Comment:
  minor fixes to make server work
```

```
**********
Label: Build.007
User: Roliver        Date:  8/20/98  Time:  4:29p
Labeled 'Build.007'
Label comment:
  True Release of Redirector/Relay/PMDB for Internal Beta.

**********
Label: Build.006
User: Roliver        Date:  8/19/98  Time:  4:21p
Labeled 'Build.006'
Label comment:
  Release for Redirector Internal RIM Beta Test.

**********  Version 178  **********
User: Roliver        Date:  8/19/98  Time:  1:43p
Checked in $/Pagemail/Redirector
Comment:
  Add support for inserting original attachments, recip lists and body
text on a reply or forward.

**********  Version 177  **********
User: Roliver        Date:  8/18/98  Time:  2:29p
Checked in $/Pagemail/Redirector
Comment:
  Revert to previous version, to remove the new debug printf's

**********  Version 176  **********
User: Roliver        Date:  8/18/98  Time:  1:28p
Checked in $/Pagemail/Redirector
Comment:
  Uncomment out several debug statements.

**********  Version 175  **********
User: Roliver        Date:  8/17/98  Time:  5:18p
Checked in $/Pagemail/PageMail
Comment:
  #ifdef out some code that is only for PERSONAL build

**********  Version 174  **********
User: Roliver        Date:  8/17/98  Time: 11:25a
Checked in $/Pagemail/Redirector
Comment:
  Change some debug log levels

**********
Label: Build.005
User: Alewis         Date:  8/14/98  Time:  3:38p
Labeled 'Build.005'
Label comment:
  (Build .004 was never officially labelled)

**********  Version 173  **********
User: Roliver        Date:  8/14/98  Time: 11:08a
Checked in $/Pagemail/Redirector
Comment:
  Add support for several new gme error codes on a mailbox->Send()
failure.

**********  Version 172  **********
User: Roliver        Date:  8/14/98  Time: 10:27a
Checked in $/Pagemail/Redirector
Comment:
```

Add several specific gme error codes, to be returned to the pager,
if a message from the pager cannot be sent out.

```
*****************  Version 171  *****************
User: Roliver          Date:  8/12/98  Time: 12:07p
Checked in $/Pagemail/Redirector
Comment:
  Don't assert() on unknown status response from relay.
Delete the pStream on a message arriving from the pager as the GME::Parameter
delete has now changed)

*****************  Version 170  *****************
User: Roliver          Date:  8/11/98  Time:  9:04p
Checked in $/Pagemail/Redirector
Comment:
  Mapi ping enabled through constructor parameter.
Fix two memory leaks.
Add change to startup to filter old unread messages. (not tested!)

*****************  Version 169  *****************
User: Jsenders         Date:  8/10/98  Time:  1:59p
Checked in $/Pagemail/Redirector
Comment:
  added reload methods to Pager and Profile interfaces for selective
reloading/refresing of the db

*****************  Version 168  *****************
User: Roliver          Date:  8/06/98  Time:  5:39p
Checked in $/Pagemail/Redirector
Comment:
  Use the gme Parameter array, instead of the linked list.
Add MORE functionality (not complete)

*****************  Version 167  *****************
User: Jsenders         Date:  8/04/98  Time: 10:49a
Checked in $/Pagemail/Redirector
Comment:


*****************  Version 166  *****************
User: Roliver          Date:  7/29/98  Time:  3:38p
Checked in $/Pagemail/Redirector
Comment:
  Change some debuglog levels.

**********************
Label: Build.003
User: Alewis           Date:  7/27/98  Time:  3:27p
Labeled 'Build.003'
Label comment:


*****************  Version 165  *****************
User: Alewis           Date:  7/27/98  Time:  2:15p
Checked in $/Pagemail/Redirector
Comment:
  Downgrade some LOG_INFORMATIONAL messages to LOG_DEBUG

**********************
Label: Build.002
User: Alewis           Date:  7/24/98  Time:  5:03p
Labeled 'Build.002'
Label comment:
```

```
****************** Version 164 ******************
User: Roliver        Date:  7/24/98  Time:  4:21p
Checked in $/Pagemail/Redirector
Comment:
  use the registry to store the lastemailaddress

****************** Version 163 ******************
User: Roliver        Date:  7/23/98  Time:  8:05p
Checked in $/Pagemail/Redirector
Comment:
  Change the welcome message.
Print the relay address, when verifying.

****************** Version 162 ******************
User: Roliver        Date:  7/23/98  Time:  5:27p
Checked in $/Pagemail/Redirector
Comment:
  Change the welcome message string.

****************** Version 161 ******************
User: Roliver        Date:  7/23/98  Time:  2:23p
Checked in $/Pagemail/Redirector
Comment:
  Only change RedirectionState to VERIFYING_ADDRESS, when in state
REDIRECTION_OFF

****************** Version 160 ******************
User: Roliver        Date:  7/23/98  Time: 12:04p
Checked in $/Pagemail/Redirector
Comment:
  Add a WelcomeMessage, but ifdef it out.
Take out the LastRelayAddress, and use the IsAddressValid() flag in the database
instead, to trigger a new verification.

****************** Version 159 ******************
User: Roliver        Date:  7/23/98  Time:  9:56a
Checked in $/Pagemail/Redirector
Comment:
  Change some Debug Levels in Printfs

****************** Version 158 ******************
User: Roliver        Date:  7/22/98  Time:  1:27p
Checked in $/Pagemail/Redirector
Comment:
  Change m_RedirectionState to tristate (OFF, VERIFYING_ADDRES, ON)
Change a lot of the debugPrintfs to make the log readable...

****************** Version 157 ******************
User: Roliver        Date:  7/21/98  Time:  6:42p
Checked in $/Pagemail/Redirector
Comment:
  Must SetLoadChanges() flag to trip the database after a
PING_RESPONSE

****************** Version 156 ******************
User: Roliver        Date:  7/21/98  Time:  4:25p
Checked in $/Pagemail/Redirector
Comment:
  Add functionality for a PING to the Relay, when the
ForwardingAddress changes (and always at startup).
```

```
****************** Version 155 ******************
User: Alewis          Date:  7/20/98  Time: 11:17a
Checked in $/Pagemail/PageMail
Comment:
  suppress handling of obsolete transport status codes

****************** Version 154 ******************
User: Roliver         Date:  7/17/98  Time:  6:29p
Checked in $/Pagemail/PageMail
Comment:
  Change some assert()s to if()s and debuglog warnings, if they can be
"handled"
Don't retrieve a Transaction (FindDeviceSendTransaction()) if it's state is
AVAILABLE
Do a check for CurrentEncryptionKey in the database on HandleDatabaseChange()

****************** Version 153 ******************
User: Roliver         Date:  7/14/98  Time:  3:14p
Checked in $/Pagemail/PageMail
Comment:
  Fix for CORPORATE version: always call NEW_MAILBOX_PACKET, and let
the conversion to NEW_MAIL take place in DoMailboxPacketProcessing() as in
PERSONAL version.

****************** Version 152 ******************
User: Roliver         Date:  7/13/98  Time:  3:53p
Checked in $/Pagemail/Redirector
Comment:
  Handle GME::SYS_CHECK by responding with GME::SYS_CHECK_OK

****************** Version 151 ******************
User: Roliver         Date:  7/13/98  Time:  2:02p
Checked in $/Pagemail/Redirector
Comment:
  Send only the MAN number in the GME::dest field - now extracted from
the database and stored locally as a string.
Handle DEVICE_DATAGRAM_ERROR like DEVICE_DATAGRAM_CONGESTED (i.e. ignore and let
MDP handle)

**********************
Label: Build.001
User: Alewis          Date:  7/09/98  Time:  5:06p
Labeled 'Build.001'
Label comment:
  First official build. Lock and Load!

****************** Version 150 ******************
User: Roliver         Date:  7/09/98  Time: 12:21p
Checked in $/Pagemail/PageMail
Comment:
  Use the CURRENT_USER registry key group.

****************** Version 149 ******************
User: Roliver         Date:  7/08/98  Time: 12:07p
Checked in $/Pagemail/Redirector
Comment:
  Don't mask the hi bit of the ETPid onto Relay.

****************** Version 148 ******************
User: Roliver         Date:  7/08/98  Time: 11:41a
Checked in $/Pagemail/Redirector
Comment:
  Allow the high bit of the refId through again.
```

```
*****************  Version 147  *****************
User: Roliver          Date:  7/08/98  Time:  9:33a
Checked in $/Pagemail/Redirector
Comment:
  Don't hardcode redirection state to true.
Take out some #pragma messages.

*****************  Version 146  *****************
User: Roliver          Date:  7/07/98  Time: 10:53p
Checked in $/Pagemail/Redirector
Comment:
  Add a general database test routine, but ifdef out.
For now, mask off the high bit on the refId to the pager, as it doesn't accept
it...


*****************  Version 145  *****************
User: Roliver          Date:  7/07/98  Time:  4:31p
Checked in $/Pagemail/Redirector
Comment:
  Properly send the transId on TRANSACTION_ERROR and MESSAGE_ERROR.

*****************  Version 144  *****************
User: Roliver          Date:  7/07/98  Time: 12:58p
Checked in $/Pagemail/Redirector
Comment:
  General clean up (move some common code into new functions).
Add refId to sent mail (from the GME)
Return TRANSACTION and MESSAGE errors to the device on failures.
Make the CORPORATE build build without errors
Temporarily stuff a 0 key into the dbase if it is not there.

*****************  Version 143  *****************
User: Roliver          Date:  7/01/98  Time:  2:28p
Checked in $/Pagemail/Redirector
Comment:
  Call PrepareMessageForPager() before converting to stream, and get
rid of some processing/formatting code directly in UserControl...

*****************  Version 142  *****************
User: Roliver          Date:  7/01/98  Time: 11:37a
Checked in $/Pagemail/PageMail
Comment:
  Send ErrorStatus back to device on some failures.
Send ErrorStatus to Relay on some failures.

*****************  Version 141  *****************
User: Roliver          Date:  6/26/98  Time:  2:32p
Checked in $/Pagemail/PageMail
Comment:
  Send Status back to Relay on success/failure of ETP.
Add filtering support and database change support.

*****************  Version 140  *****************
User: Roliver          Date:  6/23/98  Time:  3:08p
Checked in $/Pagemail/PageMail
Comment:
  comment out the filter checking for a little while...

*****************  Version 139  *****************
User: Alewis           Date:  6/23/98  Time: 10:24a
Checked in $/Pagemail/PageMail
```

```
Comment:
  RIMMessage class totally rewritten

****************** Version 138 ******************
User: Roliver          Date:  6/19/98  Time: 11:16a
Checked in $/Pagemail/PageMail
Comment:
  Fix logic bug, and ignore filter load failures on startup for now...

****************** Version 137 ******************
User: Roliver          Date:  6/19/98  Time: 10:47a
Checked in $/Pagemail/PageMail
Comment:
  printf typo

****************** Version 136 ******************
User: Roliver          Date:  6/19/98  Time: 10:46a
Checked in $/Pagemail/PageMail
Comment:
  Rename to RelayDomanName

****************** Version 135 ******************
User: Roliver          Date:  6/19/98  Time: 10:09a
Checked in $/Pagemail/PageMail
Comment:
  Add refId to varios printfs.
Change transid to use the msgId.

****************** Version 134 ******************
User: Roliver          Date:  6/18/98  Time:  1:48p
Checked in $/Pagemail/PageMail
Comment:
  Hardcode the redirection state to true for now.
Remove the RelayEmailAccount from Usercontrol/Registry

****************** Version 133 ******************
User: Cdunk            Date:  6/18/98  Time: 12:53p
Checked in $/Pagemail/PageMail

****************** Version 132 ******************
User: Alewis           Date:  6/18/98  Time: 10:23a
Checked in $/Pagemail/PageMail
Comment:
  Changed the interface from "device" to "transport". The transport
layer is now represented by a single object. Individual device objects have been
moved into the PageMail project.

****************** Version 131 ******************
User: Roliver          Date:  6/17/98  Time:  6:02p
Checked in $/Pagemail/PageMail
Comment:
  Remove format object.
Add filter object.
Filter all new_mail going to device.
Implement HandleDatabaseChange() routine that reloads filters and sets the
redirection state whenever something in the database changes...

****************** Version 130 ******************
User: Cdunk            Date:  6/16/98  Time:  5:16p
Checked in $/Pagemail/PageMail
Comment:
  For non-"personal" usercontrol calling GME encode without extra
addressing.
```

```
****************  Version 129  ****************
User: Cdunk          Date:  6/15/98  Time:  8:22p
Checked in $/Pagemail/PageMail
Comment:
  Using ETP with just one value for DATA (i.e. not direction
dependant)

****************  Version 128  ****************
User: Cdunk          Date:  6/15/98  Time:  6:08p
Checked in $/Pagemail/PageMail
Comment:
  Updated deletion of the parameters in a gme parameter list

****************  Version 127  ****************
User: Cdunk          Date:  6/15/98  Time:  6:04p
Checked in $/Pagemail/PageMail
Comment:
  Fixed incorrect memory freeing when using GME

****************  Version 126  ****************
User: Cdunk          Date:  6/15/98  Time:  5:52p
Checked in $/Pagemail/PageMail
Comment:
  uses modified gme interface

****************  Version 125  ****************
User: Roliver        Date:  6/12/98  Time:  3:40p
Checked in $/Pagemail/PageMail
Comment:
  Allow NONE and VALID_UNKNOWN_ATTACH types of email to be forwarded.

****************  Version 124  ****************
User: Roliver        Date:  6/11/98  Time:  6:21p
Checked in $/Pagemail/PageMail
Comment:
  Allow STATUS notifications for nonexistant DeviceSend transactions,
in case they have been cancelled out from under relay.

****************  Version 123  ****************
User: Roliver        Date:  6/11/98  Time:  6:17p
Checked in $/Pagemail/PageMail
Comment:
  Use ETP STATUS constants not DeviceTransaction constants.

****************  Version 122  ****************
User: Roliver        Date:  6/10/98  Time:  5:56p
Checked in $/Pagemail/PageMail
Comment:
  change __PERSONAL__ to PERSONAL, and fix ifdef build bug

****************  Version 121  ****************
User: Roliver        Date:  6/10/98  Time:  3:49p
Checked in $/Pagemail/PageMail
Comment:
  Fix delete bug for mailId

****************  Version 120  ****************
User: Cdunk          Date:  6/09/98  Time:  4:38p
Checked in $/Pagemail/PageMail
Comment:
  Using Crypto code now
```

```
****************  Version 119  ****************
User: Roliver        Date:  6/08/98  Time:  5:06p
Checked in $/Pagemail/PageMail
Comment:
  Add pipelining and remove device and link stuff for personal

****************  Version 118  ****************
User: Roliver        Date:  6/01/98  Time:  5:06p
Checked in $/Pagemail/PageMail
Comment:
  don't define __PERSONAL__ in checked in version

****************  Version 117  ****************
User: Roliver        Date:  6/01/98  Time:  4:48p
Checked in $/Pagemail/PageMail
Comment:
  Complete the integration with Craig's GME and ETP changes...

****************  Version 116  ****************
User: Roliver        Date:  6/01/98  Time:  3:48p
Checked in $/Pagemail/PageMail
Comment:
  Add in Craig's CMIME and GME changes...

****************  Version 115  ****************
User: Roliver        Date:  6/01/98  Time:  3:17p
Checked in $/Pagemail/PageMail
Comment:
  Changes for Personal Pagemail, and incorporating GME & ETP into
Pagemail.


**********************
Label: 0.0.4
User: Roliver        Date:  6/01/98  Time: 11:46a
Labeled '0.0.4'
Label comment:
  This label is pre- Personal Pagemail. GME, CMIME and ETP have
been added to the project, though not fully incorporated into usercontrol as of
this label.

****************  Version 114  ****************
User: Roliver        Date:  5/20/98  Time: 10:51a
Checked in $/Pagemail/PageMail
Comment:
  Add username do a debugprintf

****************  Version 113  ****************
User: Pagemailadmin  Date:  5/15/98  Time: 12:39p
Checked in $/Pagemail/PageMail
Comment:
  change some debug levels

****************  Version 112  ****************
User: Pagemailadmin  Date:  5/15/98  Time: 12:33p
Checked in $/Pagemail/PageMail
Comment:
  Change some debug levels to informational.

****************  Version 111  ****************
User: Roliver        Date:  5/15/98  Time: 11:16a
Checked in $/Pagemail/PageMail
Comment:
```

Fix Deadlock fix between Datagram lock and QueueAccessLock during CancelDatagram call.

```
*****************  Version 110  *****************
User: Roliver          Date:  4/30/98  Time: 12:05p
Checked in $/Pagemail/PageMail
Comment:
  Move some Printf's around.

*****************  Version 109  *****************
User: Alewis           Date:  4/29/98  Time: 10:18p
Checked in $/Pagemail/PageMail
Comment:
  Fixed memory leak in FindAndQueueAllNewMail - ppEntryId was not
being deleted if no messages found

*****************  Version 108  *****************
User: Ipatters         Date:  4/29/98  Time:  3:23p
Checked in $/Pagemail/PageMail
Comment:
  Add printf for shutting down each user.

*****************  Version 107  *****************
User: Roliver          Date:  4/29/98  Time:  3:00p
Checked in $/Pagemail/PageMail
Comment:
  Pass threadId to DoWork() for printf's
Modify some printf's

*****************  Version 106  *****************
User: Roliver          Date:  4/28/98  Time: 12:01p
Checked in $/Pagemail/PageMail
Comment:
  Remove direct call to Alarm.
Add the user name for DeviceSend status handling printf's

*****************  Version 105  *****************
User: Roliver          Date:  4/27/98  Time:  4:45p
Checked in $/Pagemail/PageMail
Comment:
  Add a call to Alarm::ActivateAlarm as a test

*****************  Version 104  *****************
User: Roliver          Date:  4/27/98  Time:  3:13p
Checked in $/Pagemail/PageMail
Comment:
  Delete the NewEntryID on moved notifications ( was a memory leak )

*****************  Version 103  *****************
User: Roliver          Date:  4/24/98  Time:  3:34p
Checked in $/Pagemail/PageMail
Comment:
  Change the slow retry time to 15 minutes.
Don't set the transferred flag after a read/moved/deleted, only after a
successful send.
Filter out new messages that have the transferred flag set.

**********************
Label: 0.0.2
User: Roliver          Date:  4/24/98  Time:  3:28p
Labeled '0.0.2'
Label comment:
  Add read/delete/moved notification support - pending
```

DeviceSends are now cancelled on this notifications.
Handle DeviceSend responses: Expired -> immediate retry, RejectedByNetwork ->
slow retry, RejectedByDevice & RejectedByPacketblaster -> cancel;
Includes Allans changes to the Utilities, after the code review.

********** Version 102 **********
User: Roliver Date: 4/23/98 Time: 5:24p
Checked in $/Pagemail/PageMail
Comment:
  Only delete the EntryId pointer array if there were EntryId's found
(in FindAllNewMessages())

********** Version 101 **********
User: Roliver Date: 4/22/98 Time: 11:57a
Checked in $/Pagemail/PageMail
Comment:
  send a copy of the CurrM2D.pMessage pointer to SendCurrMessage() as
this function modifies the pointer.

********** Version 100 **********
User: Roliver Date: 4/22/98 Time: 11:09a
Checked in $/Pagemail/PageMail
Comment:
  Add Timer to UserControl's inheritance for the slow DeviceSend
retry.

********** Version 99 **********
User: Ipatters Date: 4/22/98 Time: 10:21a
Checked in $/Pagemail/PageMail
Comment:
  Update the Queued Count on Sucessful message delivery

********** Version 98 **********
User: Ipatters Date: 4/21/98 Time: 5:01p
Checked in $/Pagemail/PageMail
Comment:
  Include the pending DeviceSend in the M2D count.
Use the proper == operator for EntryID comparison.

********** Version 97 **********
User: Ipatters Date: 4/21/98 Time: 4:30p
Checked in $/Pagemail/PageMail
Comment:
  Don't ever set the current pending message to transferred on
shutdown.
Delete the EntryId pointer array in FindAllNewMessages().
Update the QueuedM2DCount more often.

********** Version 96 **********
User: Ipatters Date: 4/21/98 Time: 4:29p
Checked in $/Pagemail/PageMail
Comment:
  Delete EntryId's for moved notification's while they're not being
handled.

********** Version 95 **********
User: Roliver Date: 4/21/98 Time: 2:03p
Checked in $/Pagemail/PageMail
Comment:
  Add Read,Delete and Move notifications (Move support not fully
implemented)
Add Mutex protection for stopping and deleting a user.

Make a much more efficient startup for new mail.
Change the DeviceSendStatus categories for better resolution in handling failures.

****
Label: Version 0.0.1
User: Roliver Date: 4/21/98 Time: 1:59p
Labeled 'Version 0.0.1'
Label comment:
Uses a Worker thread pool.
EntryId's are used to track new mail.
(Delete and Read cancelling is not yet implemented)
(Send/Receive failures are not yet handled)

********** Version 94 **********
User: Roliver Date: 4/14/98 Time: 12:58p
Checked in $/Pagemail/PageMail

********** Version 93 **********
User: Roliver Date: 4/14/98 Time: 12:48p
Checked in $/Pagemail/PageMail
Comment:
Add Read,Deleted and Moved mailbox notifications.

********** Version 92 **********
User: Roliver Date: 4/14/98 Time: 11:29a
Checked in $/Pagemail/PageMail
Comment:
Updates from after code review.

********** Version 91 **********
User: Roliver Date: 4/03/98 Time: 3:32p
Checked in $/Pagemail/PageMail
Comment:
Move the WRP queues to the Workee object. Global WorkRequestQueue now takes only DoWorkRequests (which just contain a pointer to the Workee).

********** Version 90 **********
User: Roliver Date: 4/02/98 Time: 6:25a
Checked in $/Pagemail/PageMail
Comment:
Add EntryID handling, so that we now don't toss OnNotifies if we're busy. Make the Transaction array a single variable (so that only one send can be done at a time).

NOTE: This code is not quite working yet. The WRP queing when we're busy Sending to the device must be handled differently. I'm checking in anyway - so that I have a reference before changing the queing model slightly.

********** Version 89 **********
User: Alewis Date: 3/27/98 Time: 4:51p
Checked in $/Pagemail/PageMail

********** Version 88 **********
User: Roliver Date: 3/27/98 Time: 11:22a
Checked in $/Pagemail/PageMail
Comment:
Change over to a WorkerThread pool model. UserControl no longer derives from Thread, the ThreadProc is effectively replaced with the DoWork() method.

********** Version 87 **********
User: Alewis Date: 3/10/98 Time: 3:42p

Checked in $/Pagemail/PageMail

********** Version 86 **********
User: Alewis Date: 3/09/98 Time: 10:01a
Checked in $/Pagemail/PageMail

********** Version 85 **********
User: Alewis Date: 3/06/98 Time: 4:49p
Checked in $/Pagemail/PageMail

********** Version 84 **********
User: Alewis Date: 3/05/98 Time: 4:09p
Checked in $/Pagemail/PageMail

********** Version 83 **********
User: Alewis Date: 3/05/98 Time: 1:57p
Checked in $/Pagemail/PageMail

********** Version 82 **********
User: Alewis Date: 3/04/98 Time: 11:29a
Checked in $/Pagemail/PageMail

********** Version 81 **********
User: Pagemailadmin Date: 3/03/98 Time: 4:13p
Checked in $/Pagemail/PageMail

********** Version 80 **********
User: Alewis Date: 3/03/98 Time: 12:27p
Checked in $/Pagemail/PageMail

********** Version 79 **********
User: Alewis Date: 3/03/98 Time: 9:10a
Checked in $/Pagemail/PageMail

********** Version 78 **********
User: Alewis Date: 3/02/98 Time: 7:55a
Checked in $/Pagemail/PageMail

********** Version 77 **********
User: Alewis Date: 2/27/98 Time: 5:10p
Checked in $/Pagemail/PageMail

********** Version 76 **********
User: Alewis Date: 2/25/98 Time: 10:17a
Checked in $/Pagemail/PageMail

********** Version 75 **********
User: Alewis Date: 2/19/98 Time: 1:02p
Checked in $/Pagemail/PageMail

********** Version 74 **********
User: Alewis Date: 2/18/98 Time: 1:42p
Checked in $/Pagemail/PageMail

********** Version 73 **********
User: Alewis Date: 2/16/98 Time: 4:42p
Checked in $/Pagemail/PMServe

********** Version 72 **********
User: Alewis Date: 2/10/98 Time: 11:40a
Checked in $/Pagemail/Server

********** Version 71 **********

User: Alewis Date: 2/09/98 Time: 5:30p
Checked in $/Pagemail/Server

********** Version 70 **********
User: Alewis Date: 2/06/98 Time: 3:36p
Checked in $/Pagemail/Server

********** Version 69 **********
User: Alewis Date: 2/03/98 Time: 1:35p
Checked in $/Pagemail/Server

********** Version 68 **********
User: Alewis Date: 1/29/98 Time: 5:38p
Checked in $/Pagemail/Server

********** Version 67 **********
User: Alewis Date: 1/21/98 Time: 2:14p
Checked in $/Pagemail/Server

********** Version 66 **********
User: Alewis Date: 1/13/98 Time: 5:32p
Checked in $/Pagemail/Server

********** Version 65 **********
User: Pagemailadmin Date: 11/26/97 Time: 9:17p
Checked in $/Server

********** Version 64 **********
User: Pagemailadmin Date: 11/14/97 Time: 4:26p
Checked in $/Server

********** Version 63 **********
User: Cdunk Date: 11/14/97 Time: 11:27a
Checked in $/Server

********** Version 62 **********
User: Pagemailadmin Date: 11/13/97 Time: 7:46p
Checked in $/Server

********** Version 61 **********
User: Pagemailadmin Date: 11/13/97 Time: 1:14p
Checked in $/Server

********** Version 60 **********
User: Cdunk Date: 11/13/97 Time: 10:13a
Checked in $/Server

********** Version 59 **********
User: Cdunk Date: 11/12/97 Time: 5:03p
Checked in $/Server

********** Version 58 **********
User: Pagemailadmin Date: 11/11/97 Time: 10:06a
Checked in $/Server

********** Version 57 **********
User: Cdunk Date: 11/10/97 Time: 4:54p
Checked in $/Server

********** Version 56 **********
User: Pagemailadmin Date: 11/10/97 Time: 4:30p
Checked in $/Server

********** Version 55 **********
User: Cdunk Date: 11/10/97 Time: 1:17p
Checked in $/Server

********** Version 54 **********
User: Pagemailadmin Date: 11/10/97 Time: 10:03a
Checked in $/Server

********** Version 53 **********
User: Pagemailadmin Date: 11/07/97 Time: 4:29p
Checked in $/Server

********** Version 52 **********
User: Pagemailadmin Date: 11/07/97 Time: 4:28p
Checked in $/Server

********** Version 51 **********
User: Cdunk Date: 11/07/97 Time: 2:42p
Checked in $/Server

********** Version 50 **********
User: Cdunk Date: 11/06/97 Time: 5:08p
Checked in $/Server

********** Version 49 **********
User: Pagemailadmin Date: 10/31/97 Time: 8:20a
Checked in $/Server

********** Version 48 **********
User: Pagemailadmin Date: 10/30/97 Time: 3:17p
Checked in $/Server

********** Version 47 **********
User: Pagemailadmin Date: 10/24/97 Time: 2:42p
Checked in $/Server

********** Version 46 **********
User: Pagemailadmin Date: 10/23/97 Time: 4:40p
Checked in $/Server

********** Version 45 **********
User: Pagemailadmin Date: 10/23/97 Time: 2:29p
Checked in $/Server

********** Version 44 **********
User: Pagemailadmin Date: 10/23/97 Time: 9:45a
Checked in $/Server

********** Version 43 **********
User: Hmajor Date: 10/22/97 Time: 1:25p
Checked in $/IntegrateControl

********** Version 42 **********
User: Hmajor Date: 10/21/97 Time: 5:32p
Checked in $/IntegrateControl

********** Version 41 **********
User: Hmajor Date: 10/21/97 Time: 5:24p
Checked in $/IntegrateControl

********** Version 40 **********
User: Hmajor Date: 10/21/97 Time: 4:21p
Checked in $/Server

```
****************** Version 39 ******************
User: Hmajor          Date: 10/20/97  Time:  4:07p
Checked in $/IntegrateControl

****************** Version 38 ******************
User: Pagemailadmin   Date: 10/16/97  Time: 10:05a
Checked in $/Server

****************** Version 37 ******************
User: Pagemailadmin   Date: 10/16/97  Time:  9:32a
Checked in $/Server

****************** Version 36 ******************
User: Hmajor          Date: 10/16/97  Time:  8:47a
Checked in $/IntegrateControl
Comment:


****************** Version 35 ******************
User: Pagemailadmin   Date: 10/15/97  Time:  2:11p
Checked in $/Server

****************** Version 34 ******************
User: Pagemailadmin   Date: 10/15/97  Time: 12:56p
Checked in $/Server

****************** Version 33 ******************
User: Hmajor          Date: 10/14/97  Time: 11:02p
Checked in $/IntegrateControl

****************** Version 32 ******************
User: Pagemailadmin   Date: 10/14/97  Time: 10:36p
Checked in $/IntegrateControl

****************** Version 31 ******************
User: Pagemailadmin   Date: 10/14/97  Time:  4:34p
Checked in $/IntegrateControl

****************** Version 30 ******************
User: Pagemailadmin   Date: 10/14/97  Time: 11:24a
Checked in $/IntegrateControl

****************** Version 29 ******************
User: Hmajor          Date: 10/10/97  Time:  1:03p
Checked in $/IntegrateControl

****************** Version 28 ******************
User: Cdunk           Date: 10/09/97  Time:  4:57p
Checked in $/IntegrateControl

****************** Version 27 ******************
User: Bgilhuly        Date: 10/09/97  Time: 10:39a
Checked in $/Server
Comment:
  took the lines out

****************** Version 26 ******************
User: Hmajor          Date: 10/09/97  Time:  1:10a
Checked in $/IntegrateControl

****************** Version 25 ******************
```

User: Cdunk Date: 10/08/97 Time: 9:49a
Checked in $/Server

********** Version 24 **********
User: Hmajor Date: 10/07/97 Time: 11:28p
Checked in $/IntegrateControl

********** Version 23 **********
User: Blinkert Date: 10/07/97 Time: 4:39p
Checked in $/IntegrateControl

********** Version 22 **********
User: Cdunk Date: 10/07/97 Time: 2:59p
Checked in $/Server

********** Version 21 **********
User: Hmajor Date: 10/07/97 Time: 12:53p
Checked in $/IntegrateControl

********** Version 20 **********
User: Hmajor Date: 10/07/97 Time: 12:03p
Checked in $/IntegrateControl

********** Version 19 **********
User: Hmajor Date: 10/06/97 Time: 3:37p
Checked in $/IntegrateControl

********** Version 18 **********
User: Hmajor Date: 10/06/97 Time: 1:25p
Checked in $/IntegrateControl

********** Version 17 **********
User: Cdunk Date: 10/03/97 Time: 9:15a
Checked in $/IntegrateControl

********** Version 16 **********
User: Hmajor Date: 10/02/97 Time: 5:22p
Checked in $/Server

********** Version 15 **********
User: Hmajor Date: 10/02/97 Time: 11:35a
Checked in $/Server

********** Version 14 **********
User: Hmajor Date: 10/02/97 Time: 1:43a
Checked in $/Server

********** Version 13 **********
User: Hmajor Date: 10/01/97 Time: 9:43p
Checked in $/Server

********** Version 12 **********
User: Cdunk Date: 10/01/97 Time: 5:14p
Checked in $/IntegrateControl

********** Version 11 **********
User: Hmajor Date: 10/01/97 Time: 4:27p
Checked in $/IntegrateControl

********** Version 10 **********
User: Hmajor Date: 10/01/97 Time: 2:48p
Checked in $/Server

```
****************** Version 9 ******************
User: Hmajor          Date:  9/30/97  Time:  4:17p
Checked in $/IntegrateControl

****************** Version 8 ******************
User: Hmajor          Date:  9/30/97  Time:  3:37p
Checked in $/IntegrateControl

****************** Version 7 ******************
User: Cdunk           Date:  9/30/97  Time:  1:14p
Checked in $/IntegrateControl

****************** Version 6 ******************
User: Hmajor          Date:  9/30/97  Time: 11:10a
Checked in $/Server

****************** Version 5 ******************
User: Hmajor          Date:  9/30/97  Time: 10:49a
Checked in $/IntegrateControl

****************** Version 4 ******************
User: Hmajor          Date:  9/30/97  Time: 10:43a
Checked in $/Server

****************** Version 3 ******************
User: Hmajor          Date:  9/26/97  Time:  5:40p
Checked in $/Server

****************** Version 2 ******************
User: Hmajor          Date:  9/25/97  Time:  5:25p
Checked in $/Server

****************** Version 1 ******************
User: Hmajor          Date:  9/25/97  Time:  5:12p
Created usercontrol.cpp
```